

PCT

世界知的所有権機関
国際事務局
特許協力条約に基づいて公開された国際出願



(51) 国際特許分類6 G06F 9/06	A1	(11) 国際公開番号 WO97/16784 (43) 国際公開日 1997年5月9日 (09.05.97)
(21) 国際出願番号 PCT/JP96/03183 (22) 国際出願日 1996年10月30日 (30.10.96) (30) 優先権データ 特願平7/281794 1995年10月30日 (30.10.95) JP (71) 出願人: および (72) 発明者 根来文生(NEGORO, Fumio)[JP/JP] 〒248 神奈川県鎌倉市十二所967-64 Kanagawa, (JP) (74) 代理人 弁理士 須山佐一(SUYAMA, Saichi) 〒101 東京都千代田区神田多町2丁目1番地 神田東山ビル Tokyo, (JP)		(81) 指定国 CA, CN, JP, KR, NZ, RU, SE, SG, US, 欧州特許 (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). 添付公開書類 国際調査報告書
<p>(54) Title: SOFTWARE PRODUCTION METHOD, PROCESSOR AND RECORDING MEDIUM</p> <p>(54) 発明の名称 ソフトウェアの生産方法、処理装置及び記録媒体</p> <p>(57) Abstract</p> <p>First, a defining member identifier constituting a screen required for software to be produced is determined. The words contained in the defining member identifier are then taken out, and a processing path diagram in which all pallets required for the software are arranged on the basis of the defining member identifier and along a flow of a process is prepared. A required file is then determined on the basis of the extracted words and prepared processing path diagram. A first base logic for editing screens or files for all of the extracted words, a second base logic for determining a processing path and a third base logic for updating a file are then prepared. Three kinds of pallets in which the first to third base logics are bound on a screen basis and pallet functions for executing these base logics autonomously and significantly in the pallets are then prepared. Then, a screen based on the pallet function concerning the first base logic is transmitted, and, after the screen has been received, the pallet function concerning the second base logic is executed. On the basis of the results of the execution of this function, the above-mentioned three kinds of pallet functions are incorporated in a pallet chain function of a structure determining one processing path out of a plurality of processing paths at least one of which comprises the execution of a pallet function concerning the third basic logic.</p> <pre>graph TD; A[定義体識別子の決定] --> B[単語の決定]; A --> C[処理経路図の作成]; B --> D[ファイルの決定]; C --> D; D --> E[基底論理の作成]; E --> F[パレット関数の作成]; F --> G[パレット連鎖関数への組込み];</pre> <p>101 ... Determine a defining member identifier 102 ... Determine words 103 ... Prepare a processing path diagram 104 ... Determine a file 105 ... Prepare base logics 106 ... Prepare pallet functions 107 ... Incorporate pallet functions in a pallet chain function</p>		

(57) 要約

まず生産するソフトウェアに必要な画面である定義体識別子を決定する。次に、定義体識別子に存在する単語を抜き出すと共に、定義体識別子に基づきソフトウェアに必要な全てのパレットを処理の流れに沿って配置した処理経路図を作成する。次に、抜き出した単語及び作成した処理経路図に基づき必要なファイルを決定する。次に、抜き出した全ての単語に対し、画面編集またはファイル編集を行う第1の基底論理、処理経路を決定する第2の基底論理及びファイル更新を行う第3の基底論理を作成する。次に、画面単位で前記第1～第3の基底論理をそれぞれ括ってなる3種のパレットを作成すると共に、各パレット内で各基底論理を自律的に有意性をもって実行させるパレット関数を作成する。そして、第1の基底論理に関するパレット関数に基づく画面を送信し、画面を受信して第2の基底論理に関するパレット関数を実行し、この実行結果に基づき第3の基底論理に関するパレット関数の実行を少なくとも1つの処理経路とする複数の処理経路から1つの処理経路を決定する構造のパレット連鎖関数に、上記の3種のパレット関数を組み込む。

情報としての用途のみ

PCTに基づいて公開される国際出願をパンフレット第一頁にPCT加盟国を同定するために使用されるコード

AL	アルバニア	EE	エストニア	LR	リベリア	RU	ロシア連邦
AM	アルメニア	ES	スペイン	LS	レソト	SD	スー・ダーン
AT	オーストリア	FR	フランス	LT	リトアニア	SG	スウェーデン
AU	オーストラリア	GA	ガボン	LU	ルクセンブルグ	SI	シングапур
AZ	アゼルバイジャン	GB	イギリス	LV	ラトヴィア	SK	スロヴァキア共和国
BB	ベルバドス	GE	グルジア	MC	モナコ	SN	セネガル
BEE	ベルギー	GH	ガーナ	MD	モルドバ	SZ	スエズランド
BFF	ブルガーニ・ファソ	GN	ギニア	MG	マダガスカル	TG	チャード
BG	ブルガリア	GR	ギリシャ	MK	マケドニア旧ユーゴスラ	TJ	タジキスタン
BJR	ベナン	HUE	ハンガリー	ML	マリ	TM	トルクメニスタン
BR	ブラジル	IEST	アイスランド	MN	モンゴル	TR	トリニダード・トバゴ
BY	ベラルーシ	ITP	イタリー	MR	モーリタニア	TT	ウクライナ
CA	カナダ	JPE	日本	MW	マラウイ	UG	ウガンダ
CF	中央アフリカ共和国	KE	ケニア	MX	メキシコ	UA	米国
CG	コンゴー	KGG	キルギスタン	NE	ニジニノヴォgorod	UZ	ウズベキスタン共和国
CH	イス	KGP	朝鮮民主主義人民共和国	NL	オランダ	VN	ヴィエトナム
CI	コート・ジボアール	KR	大韓民国	NO	ノルウェー	YU	ユーゴスラビア
CM	カメルーン	KZ	カザフスタン	NZ	ニュージーランド		
CN	中国	LK	リビエンシタイン	P	ポルトガル		
CZ	チエコ共和国		スリランカ	RO	ルーマニア		
DE	ドイツ						
DK	デンマーク						

明細書

ソフトウェアの生産方法、処理装置及び記録媒体

技術分野

この発明は、業務用ソフトウェアやゲーム用ソフトウェア、その他あらゆる分野のソフトウェアに適用されるソフトウェアの生産方法、処理装置及び記録媒体に関する。

背景技術

どれだけOSやパッケージが良く作られても、それだけでは利用者の要求を満たすことができない。ソフトにはユーザーが自分で開発しなければならない論理部分が必ず存在するからである。重要なことはその部分についてソフトの生産技術が適用されなければならないことである。即ち、生産技術とはソフトの短期開発、保守課題からの解放、上流仕様とプログラム仕様を連接する方法、プログラムの機能品質を保証する等の効果が發揮されるものでなければならない。

従来方法は要件機能を抽出してソフト化する機能分割法であるが、機能分割法から付加的なロジックが派生することは殆どの場で理解されていない。そして、この思考方法には基本的な欠陥があり、すべてのソフト課題はここから生じている言っても過言でない。この問題を基本的に改善するにはこの思考方法に替わる新しい概念、即ち、要件事象を封じ込める概念分割法の理論が必要である。

本発明者は、かかる概念分割法の理論を案出し、これをLYEE (GOVERNMENTAL METHODOLOGY for SOFTWARE PROVidence)と命名した。本理論は将来に向けた新たな思考方法の基準を与えるものである。

ソフトの生産保守性の非効率性はソフト構築の手段をソフトの特質を無視する機械機能的なパッケージ化に向かわせている。このソフト自体もまた非正解型であり、且つ、非生産的なままで開発並びに保守が行われているが、結果的に利用

者の利用環境は次第に信頼性、可用性、適用性（R A S）機能の局面で悪化の一途をたどっている。この時期必要なことは生産の為の基盤整備である。

この分野においては、生産技術の問題に手をつけずにいながら、即ち、プログラムの単位的な処理機能を品質的に管理する方法を持たず、上流だけで良質のソフトが決定できると主張するのである。異常と言わねばならない所以である。ソフトの生産方法は上流下流に分けて行われるような性質のものではなく、理論的に捉える所から始めなければならない。L Y E E 理論は正にこれを満たすものである。

即ち、本発明の目的は以下の通りである。

- (1) ソフト開発者及び保守者の生産能力を向上させること。
- (2) 上流情報と下流情報を理論的に連接させること。
- (3) 単に業務ソフトの開発に限らず、O S、ミドルソフト、ゲームソフト等幅広いソフト分野の開発に適用できるプログラム構造を提供すること。
- (4) 上流情報を最小化しソースプログラムを生成すること。

発明の開示

以下、本発明者の論文

「題目：ソフトの存在由来

副題目：ソフトの真相構造

副題目：三次元思考方法を二次元思考方法から誘導する理論」

の一部を抜粋する。

存在には真相と実相がある。

其れらは様相の連鎖で律されている。

真相とは存在由来のことであり、其の連鎖は連想して成立由来を表す連鎖となる。

実相とは成立由来の連鎖が重複して表す事象の事である。

生命体も其の思考方法も存在は全てが実相に外ならない。

実相に至る二つの連鎖と重複連鎖を総称して実相連鎖、其の連鎖を成立させる作用を論理化作用と呼ぶ。

生命体が自覚する存在とは、例えば、水がH₂Oであると言う様な実相のことで、ついてだけであり、水の真相が自覚できる由ではない。

ニュートンもAINシュタインも自覚したのは実相であり、譬え、二人を統合したとしても其の自然則の真相が解明される由ではない。

様相は固有の時間を持って存在し、その値は同じではない。

そして、時間速度が異なる様相は同じ空間に占めることはできないので、其の占有する空間は異なる事になる。

真相を表す様相の時間速度は存在の仕方として実相を表す様相の時間速度

よりも先行して成立しなければならないので、

真相を表す連鎖は認識空間と呼ぶ領域に、実相を表す連鎖は認識空間と呼ぶ時間速度が異なる領域に展開する。

記憶とは生命体が創造する認識空間の部分集合である。其処には認識空間の様相が存在する。生命体はこの記憶を用いて更なる連鎖を創造し、既存の実相連鎖との間で同化を図る。同化とは二つの連鎖の対応関係の成否を図る作用の事である。同化作用が成立すると其の実相連鎖は記憶として受容される。

生命体も其れ自身固有の時間速度を持つ認識空間に存在する実相の事である。其れ故、生命体が創造する連鎖も実相を表す連鎖となる。そして、其の連鎖の様相は記憶として存在する様相に限られるものとなる。換言すれば、生命体は時間速度の異なる様相を連鎖する事は原理的に不可能なのである。

其れ故、生命体が記憶する様相は生命体が創造できる様相に限られ、真相を表す様相は受容されることはないのである。其れ故、生命体の自覚とは生命体が創造する実相で有る限り、記憶されていない真相が自覚として現象化される事はないのである。

生命体が創造する様相は其の稠密の進展と共に生命体の物理性に有効となり、結果的に、其の様相に支配を委ねる形になった。

換言すれば、生命体は真相よりも実相による支配を選択した由である。

もし、生命体が真相を自覚できる思考方法に開眼するならば、其の進展は実相効果の比ではないはずである。

本論文では真相を理論的に求める思考方法を論じるものである。

其の目的は真相の構造がソフト課題を克服する究極のソフト構造となるからである。

第一章

1. 1 論理方程式

様相は意味性・有意性・時間速度を構造要素とする独立した構造体である。

意味性とは存在を表す為の意志である。

この意志は出現順次で決定される。

出現順次とこの意志は唯一の対を成し、其れ故同質様相が現れる事はない。

後述する論理方程式の連鎖順次はこの出現順次を表すものである。

そして、この意志は空間的広がりを表すものとなる。

有意性とは意味性に共棲し意味性の存在性を拡大させる為の意志である。

意味性はこの有意性に基づいて他の様相の意味性と連鎖する。後述する連鎖はこの意志の作用で誘導されるものである。

有意性の拡大作用は普遍的である。

本理論ではこの有意性を論理化関数として表す。

時間速度は様相に帯同し其の様相の固有する時間速度によって存在する空間を決定する。様相が存在する空間域の事を母集団と呼ぶ。意識空間、認識空間とはこの母集団の事である。

意味性は其の存在空間する空間域によって有り様を変える。

即ち、意識空間では大きさを持つ一様な空間的広がりとして存在する。

認識空間では多様で多重的な空間的広がりとして存在する。

時間速度は後述する。

連鎖は決定された母集団の部分集合の様相の空間的広がりを大きさの降順に並べたものである。この順序列を降順順序列と呼ぶ。

この部分集合は母集団に属するひとつ以上の様相から成るものである。

様相の有意性は連鎖されて単位化される。其の事で部分集合化された様相の空間的広がりの総和や重複の関係を成立する。

総和に最も近似する様相が母集団に存在すれば、其れを連鎖に対し等価な様相と呼ぶ。

総和、等価、重複は後述する。

連鎖は連鎖と等価な様相の出現順次を現す為のものである。そして、論理方程式はこの関係を正規化したものである。そして、この論理方程式を用いて連鎖に関する更なる有り様を推論するために利用する。

連鎖と等価な様相が唯ひとつである場合、其の連鎖と連鎖の様相は唯一的である。

複数個存在すれば多義的である事を表す。

もし、存在しなければ其她们は全て不定となる。

意識空間に存在する様相は其の固有する時間速度から独立である。其れ故、この様相は重複して存在することはない。この様相の事を意味要素と呼ぶ。

当然、其の連鎖も重複はしない。

其れ故、其の連鎖も意識空間に於いてだけ成立する事になる。結果的に、意識空間は唯ひとつの空間として律することが出来る。

他方、認識空間に存在する様相は其の固有する時間速度から独立して存在する事とは出来ない。其れ故、この様相は重複して存在する。この様相の事を状態要素と呼ぶ。

当然、其の連鎖も重複する。

其れ故、其の連鎖も認識空間に於いてだけ成立する事になる。結果的に認識空間は重複される事に於いて、意識空間のように唯ひとつの空間ではなく、其の部分空間は多義的となり、全体は不定となる。

重複とは其の様相の空間的広がりが多重的に拡大される事である。

連鎖は更なる連鎖を誘導する。この関係に於いて、新たな連鎖と等価な様相を帰結様相、新たな連鎖を誘導する為に媒介となる連鎖に等価な様相を媒介様相と呼ぶ。

意識空間で成立する連鎖を意識連鎖、認識空間で成立する連鎖を転位連鎖と呼ぶ。意識連鎖は存在の真相、即ち、存在由来を表し、転位連鎖は存在の実相、即ち、成立由来を表す。例えば、水が様々な実相を表すのは水を表す転位連鎖の重複によるものである。そして、重複した転位連鎖を状態連鎖と呼ぶ。

転位連鎖を成立させる為には意識連鎖と転位連鎖を連想させる事が必要である。

しかし、この二つの連鎖が存在する空間は同じではない為、時間速度が異なり、直接連想を果たす事が出来ない。この理由から意識空間には意識連鎖と連想する相対連鎖が必要になる。相対連鎖は重複が出来ない意識連鎖を重複可能にする為のものである。即ち、意識連鎖はある時間の間だけ最も低位の重複を成立させる相対連鎖を現させる。この時間を転位時間と呼ぶ。転位時間は対象となる意識空間に存在する様相の時間速度と連想される認識空間の様相の時間速度の比率で決まる時間の事である。

意識連鎖は相対連鎖を介して転位連鎖と連想する。

唯一的な様相は悠久に存在するが、多義的な様相は滅する。しかし、不定でなければ意味要素と同化する可能性を持つ。意味要素と同化する様相を回帰様相と呼ぶ。回帰様相は意味要素と同様に悠久に存在する事が出来る。不定となる様相は唯必滅するだけである。

既述した様に連鎖は様相の有意性の作用から決定されるものである。

其れは、母集団の部分集合、部分集合に属する様相の降順順序列、そして、其の降順順序列と等価な様相を母集団から決定する事である。

有意性の作用を論理化関数とする。

以下に其のアルゴリズムを示す。

①媒介様相に属する様相それぞれについて近似する様相を母集団から抽出する。
近似する様相とは其の様相の意味性が表す空間的広がりの大きさが近似する事である。

近似には+側と-側の空間的広がりの大きさがあるので、其の両方を抽出する。
土の様相がひとつとなる場合には其のひとつの様相が抽出されるが、更に、其のひとつの抽出した様相に対して、更に近似する二つの様相を求めなければならない。
二つの様相が抽出される迄この作用は継続される。

抽出する様相が存在しなければこの作用は其処迄とする。

抽出する様相が多義的に存在すれ場合には其のどれかを任意的に抽出する。

②抽出した全ての様相と媒介様相の様相を加えて新たな連鎖の様相の集合とする。
集合に属する様相が異なる母集団を持つ場合、
様相の数の大きな方の母集団とその様相が残り、様相の数の小さな方の母集団と
其の様相は排除される。

③同じ様相が存在すれば唯ひとつに正規化される。

④新たな集合の様相が表す空間的ひろがりの大きさを降順に並べる。これを降順
順序列と呼ぶ。

⑤降順順序列と等価な様相を母集団から決定する。

等価な様相とは総和関係で算定する

連鎖が空間的広がりの大きさに+側で近似する様相の事である。

総和関係とは

降順順序列の様相順位が奇数順位にある様相の空間的広がりを

様相順位の一番目の様相の空間的広がりの中に吸収し、

偶数順位にある様相の空間的広がりを様相順位の一番目の様相の空間的広がりに
算術的に加えて空間的ひろがりの大きさを決定する事である。

(1) 論理方程式

$$E_{k,R} = \Phi_{k-1} (E_{k-1,R} + (E_m)_D)$$

論理方程式は媒介様相 $E_{k-1,R}$ とその母集団 $(E_m)_D$ の様相に作用する論理化
関数 Φ_{k-1} から誘導される帰結様相 $E_{k,R}$ の関係を表す。

添え字 K は連鎖が繰り返される場合の順次で連鎖順次を表す。

連鎖順次は 1 ≤ K の自然数として規約される。

本理論では意識空間の単位時間を連鎖から連鎖が誘導される時間と定義して捉えるので、連鎖順次は意識空間の単位時間を表すものとなる。
連鎖順次は重複する事なく常に唯一である。

添え字 R は連鎖が重複する場合の順次で重複順次を表す。

重複順次は $0 \leq R$ の自然数として規約される。

本理論では認識空間の単位時間を重複から重複が誘導される時間の事と定義して捉えるので、重複順次は認識空間の単位時間を表すものとなる。
重複順次は多義的である。

即ち、認識空間では同じ実相が異なる重複順次で出現する事になる。

連鎖順次と重複順次が同時に変わる事はない。

連鎖順次が変わる場合は重複順次は $R = 0$ で固定され、

重複順次が変わる場合は連鎖順次が固定される。

媒介様相に作用する論理化関数 Φ_{k-1} を言い換えれば、連鎖順次 $K - 1$ の連鎖に作用する論理化関数の事になる。

母集団 $\{E_m\}_D$ の添え字 D は意識空間の場合 W、認識空間の場合 C で表される。

帰結様相と媒介様相の部分集合の様相の数は $N_{k+1} \geq N_k$ の関係である。

論理方程式は以下の様に展開する事が出来る。即ち、

$$E_{k,R} = \Phi_{k-1} (E_{k-1,R} + \{E_m\}_D)$$

$$E_{k-1,R} = \Phi_{k-2} (E_{k-2,R} + \{E_m\}_D)$$

(2) 実相連鎖

実相連鎖は 3 種の連鎖に連想を成立させ且つ其のひとつである転位連鎖に重複を成立させる事である。本理論では実相連鎖を成立させること、即ち、論理化作用

を認識的に行う事を解法と呼ぶ。解法については後述する。

ここでは実相連鎖を表す全ての連鎖を論理方程式で説明する。

0 1. 意識連鎖 ($K \geq 1, R = 0$)

意識連鎖は意識空間で成立する意味用を連鎖様相とする真相を表す連鎖の事である。

即ち、

$$\begin{aligned} M_{k,0} &= \Phi_{k-1} (M_{k-1,0} + \{M_m\}_w) \\ &= \Phi_{k-1} \Phi_{k-2} \cdots \Phi_2 \Phi_1 \Phi_0 (M_{0,0} + \{M_m\}_w) \end{aligned}$$

論理化関数は唯ひとつであるので、其れを Φ で表せば、上式の論理化関数列は Φ^k である。故に、上式を以下の様に表す。即ち、

$$M_{k,0} = \Phi^k (M_{0,0} + \{M_m\}_w)$$

ここで、 $M_{0,0}$ を境界条件と呼ぶ。

意識連鎖がソフト化される場合、

境界条件は定義体に置換される。

意味要素の数を M とすれば、意識連鎖の成立数 G_M は以下の通りである。

即ち、 $G_M = {}_M C_1 + {}_M C_2 + \cdots + {}_{M-1} C_{N-1} + {}_M C_N$ 、記号 C は組合せ計算式を表す。

最大の意識連鎖とは意識空間の全ての意味要素からなる連鎖である。

意識連鎖がこの数に到達すると意識空間に於ける論理化作用は停止する。

0 2. 相対連鎖 ($K \geq 1, R = 1$)

意識連鎖を転位連鎖に連想する為に意識連鎖から誘導される意識空間で成立する連鎖の事である。即ち、

$$M_{k,1} = \Phi^K (M_{0,1} + \{M_m\}_w)$$

0 3. 転位連鎖 (K ≥ 1, R = 1)

相対連鎖から連想される意識空間に存在する状態要素を連鎖様相とする認識空間で成立する成立由来の連鎖の事である。

連想の際、母集団が意識空間から認識空間に置換される。即ち、

$$S_{k,1} = \Phi^K (M_{0,1} + \{S_m\}_c)$$

生命体は転位連鎖の連想が転位時間内に終了させられない場合、この連想は廃棄されるか、或いは、記憶情報を用いて代替的に創造するかのいずれかを選択する。しかし、創造される転位連鎖は論理化作用の摂理に反するものである事は明らかである。この観点から見れば、生命体の創造とは多義的である故に偽性であり、換言すれば、

人類の文明とは偽性の限界を目指すものと見る事が出来る。

0 4. 状態連鎖 (K = 固定、R ≥ 2) の論理方程式、

転位連鎖の重複から連想される実相を表す連鎖の事である。

$$S_{k,R} = \Phi^K (M_{0,1} + \{S_m\}_c)^R$$

其れ故、状態連鎖は其の重複度合いで其の帰結様相は様々な実相を現象化させる。

重複順次 R は決定された転位連鎖に作用する。

状態連鎖の構造は他の連鎖と異なり、

転位連鎖に属する様相の空間的広がりが多重化される事である。

例えば、水の成立様相が様々であるのは転位連鎖の重複度合によるものである。

生命体が創造する状態論理の様相は、自覚・感情・言葉・文章・態度・行動として現象化される。

0 5. 回帰論理（略）

（3）論理化作用

論理化作用は意識連鎖、相対連鎖、並びに、転位連鎖と其の重複である状態連鎖を連想させる事である。

転位連鎖から相対連鎖を誘導する論理方程式を以下に示す。

$$\text{即ち、 } M_{0,1} = \Phi^{-K} (S_{k,1} + \{M_m\}_w) \quad ①$$

転位連鎖と状態連鎖は以下の関係にある。即ち、

$$S_{k,R} = (S_{k,1})^R \quad ②$$

$$\text{故に、 } S_{k,1} = (S_{k,R})^{-R} \quad ③$$

式③を式①に代入して式①を改める。即ち、

$$M_{0,1} = \Phi^{-K} ((S_{k,R})^{-R} + \{M_m\}_c) \quad ④$$

式④は生命体は状態連鎖しか自覚する事が出来ないので、式③で重複順次 R で決定される状態連鎖から転位連鎖を求め、其の転位連鎖から意識連鎖を求める為の論理方程式である。式①、式②で示される逆論理化関数は論理化関数が普遍的なので必ず成立し求める事が出来る。

其の結果、意識空間の様相を仮定する事が出来れば、其処に存在する意識連鎖を求める事が出来る。式④を逆論理方程式と呼ぶ。
解法とはこの作用の事に外ならない。

1. 2 時間摂理（図 2 3 参照）

意識論理が決定される時間を意識論理の単位時間とする。
転位時間が重複される時間を認識空間の単位時間とする。
意識空間に於ける連鎖順次は唯一的である。
認識空間は部分的に成立し全体は不定であるので、認識空間に於ける重複順次は部分空間ごとに定まり多義的となる。

単位時間が様相の個別的な事情で相違する事を付度して、絶対時間で捉えた様相の固有時間を単位時間の逆数で表し時間速度と呼ぶ。

状態連鎖は同化して生命体の記憶となり、其の記憶が感情・言葉・行動・文書などに現象化される。しかし、其の以前に
其の発端となる意識連鎖は成立して存在しなければならない。
即ち、意識連鎖が成立する時間速度は状態連鎖が成立する時間速度よりもおおきくなければならないのは必然である。

単位時間、時間速度、転位時間、回帰時間を総称して時間摂理と呼ぶ。以下に、定義を記す。

（1）単位時間

意識連鎖を決定する単位時間 : $t_U (M_{k,0})$

状態連鎖を決定する単位時間 : $t_U (S_{k,R})$

二つの空間の単位時間の関係 : $t_U(M_{k,0}) \ll t_U(S_{k,R})$

(2) 時間速度

意識連鎖の時間速度 : $t_V(M_{k,0}) = 1/t_U(M_{k,0})$

状態連鎖の時間速度 : $t_V(S_{k,R}) = 1/t_U(S_{k,R})$

二つの空間の時間速度の関係 : $t_V(M_{k,0}) >> t_V(S_{k,R})$

時間速度の大きな様相は独立して存在する。

時間速度の小さな様相は独立して存在する事が出来ない。

(3) 転位時間

認識空間での転位時間 : $t_T = t_V(M_{k,0}) / t_V(S_{k,R})$

(4) 回帰時間（略）

第二章 ソフト化解法

意識連鎖を誘導する逆論理方程式、即ち、 $M_{0,1} = \Phi^{-K}((S_{k,R})^{-R} + \{M_m\}_c)$
の解法は特定された認識事象（状態連鎖）に関して成立する。

本章では特定された認識事象を電算機で稼働するソフトとして解法する場合について述べる。そして、この解法論の事をソフト化解法と呼ぶ。

ソフト化解法で書き改められる逆理論方程式をシナリオ関数と呼ぶ。

シナリオ関数の説明は第三章で行う。

シナリオ関数はソフトの唯一的な構造を現す事に帰結する。

其れ故、本理論ではソフト開発作業はこのシナリオ関数を満たす作業の事に帰結する。

換言すれば、ソフト化解法の過程に現れる条件を具象化する事と同質の作業となる。

其れは解法過程に現れる条件が決定されるべき作業の厳密な命題を捉える事になっているからである。

通常、思考方法の中で定型化できる思考方法を総称して方法論と言い換えるのである。

本理論の場合、具象化する為の思考方法は

其の命題自体が経験的知識と言ったものではなく解法と言う論理的過程の中で捉えられたものなので、従来に比べ必然的に高割合で定型化される事になる。

第三章 開発方法論

ソフト化解法を開発方法論化する事について述べる。

従来のソフトとは要件事象、即ち、状態連鎖を S E の属人的能力（経験、知識、応用力）で前以て機能的に規約し其れをプログラム化するものである。

この場合の電算機は其のプログラムから S E が前以て既知とした状態連鎖を短時間で情報的に再現する役割を担うものである。

即ち、従来のプログラムは電算機を通じて短時間で既知なる状態を再現する事にあり、新たな状態連鎖を生み出す由ではない。

其れ故、新たな状態連鎖は再現された情報が衝動となって行われる生命体の論理化作用に委ねるしかないものである。

本理論を開発方法論化するのは意識連鎖を表すプログラムを紙上（認識的）に作り出す為である。そして、其のプログラムを電算機で構造的に重複させる事によって、其の意識連鎖に連想する状態連鎖を電算機から生み出せると言うものである。

所謂、要件事象とは状態連鎖の事であり、其れは常に多義的（CHAOS）に存在する。其れ故、繰り返し（馴れ）による自己帰結は同化され易い傾向を表すが、成立由来の重複連鎖を創造する事を前提するに於いて結果的に非生産的なものとなる所に根本的な欠陥があった。

其れ故、要件事象の状態連鎖の成立は事前に規約するしか方法はなく、電算機に其れを委ねると言う事は基本的に不可能なことであった。例えば、演算的に其れを誘導すると言う発想はAIと呼ばれるもので代表されるが、単に商業的な希望的着想の域を越えられるものではない。本理論はこの問題を電算機で代替可能とさせる誘導方法を表すものである。

本理論から導出された意識連鎖を捉えるプログラムの構造は状態連鎖を捉える構造に較べ著しく簡易化されたものとなる。其の結果、意識連鎖を作り出すソフトを決定する人為的作業も必然的に簡易化される。其の結果、ソフトの開発・保守の作業局面に著しい改善的效果をもたらす。其の効果の要点を以下に掲げる。

- ①従来方法では不可欠となっていたSE作業の50%は不要となる。
- ②関係者の論理化作用を大幅に削減する事が出来る。
即ち、状態連鎖を捉える従来の人為的作業では其の内の10%程度しか機械化する事が出来ない。90%が作業者個人の能力に委ねられざるを得なかった。意識連鎖を捉える本理論の場合では其の内の70%を機械化する事が出来る。
- ③従来方法でのソースプログラムの論理部分は20%迄に圧縮される。
- ④従来方法では其の上流工程と下流工程がプログラマーの自己的な論理化作用で連想されるしか手段がなかったので、上流工程と下流工程の関係は当事者以外に

は分かりにくく、其の為、保守作業の生産性は開発作業に較べ更に十分の一程度に低下していた。本理論の場合、上流と下流の関係が理論的に連想されるので、この部分のプログラマーの個人的な論理化作用は不要となる。

換言すれば、其の関係は個人的な心因に埋没したものでなくなり、本理論の理論的構造を理解すれば誰もが同じ論理化作用を行う事が出来る様になる。其の結果、保守作業の生産性は飛躍的に改善される事になる。

(01) シナリオ関数

後述する空間モデルを用いてソフト化解法された逆論理方程式はソフトの意識連鎖の構造を表す。意識連鎖の構造は普遍的に唯ひとつとなるので、この場合のソフト構造も必然的に唯ひとつとなる。これをシナリオ関数と呼ぶ。

画面を基底定義体とするシナリオ関数 T_0 を以下に示す。即ち、

$$\begin{aligned} T_0 = & \Phi_0 (\{\Phi_{p,k} (L_{i,2}, T_{1,f})\} + \{\Phi_{p,k} (L_{i,3}, T_{1,g})\}) \\ & + \{\Phi_{p,q} (L_{i,4}, T_{1,q})\}) \end{aligned}$$

ここで、

Φ_0 は基底定義体を画面とするパレット連鎖関数を表す。

$\Phi_{p,k}$ はパレット関数を表す。添え字 k は画面識別子を表す。

$L_{i,2}$ は単語 i の意味領域 $W02$ の基底論理を表す。

$L_{i,3}$ は単語 i の意味領域 $W03$ の基底論理を表す。

$L_{i,4}$ は単語 i の意味領域 $W04$ の基底論理を表す。

$W02$ 、 $W04$ の単語 i は画面 k に存在する単語である。

$W03$ の単語 i はシステムに存在する全単語である。

T_1 は基底定義体をファイルとするシナリオ関数で、従来型プログラムを連想する。其の添え字 f 、 g 、 q はファイル識別子を表す。

即ち、 T_1 もまた基底定義体をファイルとするシナリオ関数である。この場合の

バレット連鎖関数 Φ_1 である。即ち、

$$T_{1,f} = \Phi_1 (\{\Phi_{p,k}(L_{j,2})\} + \{\Phi_{p,k}(L_{j,3})\} + \{\Phi_{p,k}(L_{j,4})\})$$

ここで、

$L_{j,2}$ は単語 j の意味領域 $W_0 2$ の基底論理を表す。

$L_{j,3}$ は単語 j の意味領域 $W_0 3$ の基底論理を表す。

$L_{j,4}$ は単語 j の意味領域 $W_0 4$ の基底論理を表す。

$W_0 2, W_0 4$ の単語 j はファイル f に存在する単語である。

$W_0 3$ の単語 j はシステムに存在する全単語である。

(2) 空間モデル (図 24、25、26 参照)

ソフトとは存在の実相と真相の関係を表す関係の事である。

ソフトの開発とは実相連鎖を成立させる事にあるが、理想を述べれば、意識連鎖を成立させる事に帰着する。しかし、意識連鎖は永久に不明であり、其の意識連鎖を成立させるには状態連鎖から意識連鎖を誘導する逆連想を成立させる事が必要である。

其の不可欠な条件は認識空間と意識空間の構造的な成立性を唯ひとつの空間で表す事である。本理論ではこの空間を空間モデルと呼ぶ。

空間モデルはソフト化解法の過程で現れる解法条件から誘導されなければならない。其れ故、ソフトは空間モデルに帰着し、

ソフト開発はこの空間モデルを決定する作業と同義となる。

空間モデルの構造について以下に説明する。

解法条件が現す認識空間の律性は受容、論理、相対に仕分される。この律性を3種の二次元領域に置き換える。そして、これを意味領域と総称する。

意味領域はW02、W03、W04で記され、
W02では受容律性、
W03では論理律性、
W04では相対律性を表すものとする。

他方、意識空間の律性は意味領域と逆連想の関係にある三次元の空間概念に置き換えられる。即ち、空間モデルとは意味領域の連鎖で認識空間を表し、其の意味領域が構成する三次元空間で認識空間を表す構造体の事である。そして、

本理論で言う意味要素は認識的に実現する事が出来ないので、開発方法論化される場合、意味要素は単語で代替するものとする。

即ち、単語は意味領域が構成する三次元空間に配される。

そして、単語は意味領域に転位（ベクトル分解）され、それぞれの律性で其の意味性を基底論理（プログラム）に置換する。基底論理は後述する。

因に、従来のプログラムは要件事象の意味性が経験や知識で仕様化され、其れが論理化された。基底論理は意味領域のそれぞれの律性が経験や知識に代わって単語の意味性を基底論理に置換する仕様となる。

この基底論理を領域的に連鎖させると認識的な状態な要件事象と等価な機能的様相となる。既述して来た連想とは単語の意味性と其の基底論理が領域的に連鎖して出現させる機能的様相の対応関係が成立する事である。

領域的な連鎖とは意味領域に順序性を与える事で、理論上、其の形式は3種類となる。

しかし、開発方法論化した場合、電算機の方式的な事情から5形式となる。この順序性は後述する処理経路図として表される。

そして、この順序性は実相の要件事象から真相を誘導する為の逆連想の成立条件

を規約するものとなる。

意味領域の律性は解法条件として既知であり、
システムに存在する単語を意味要素に代替させる事で、基底論理を満たし、
基底論理が存在する意味領域の領域的連鎖は3形態を基にすれば、其の組み合わ
せて如何なる様相でも全て律する事が出来るので、其れは既知と同値である。
そして、この空間モデルの構造から
意識連鎖を誘導する為の逆連想を成立させる事が出来る。

(3) 基底論理 (図27参照)

システムに存在する単語を意識空間に存在する意味要素に置き換えるものとする。
意味要素が其の有意性から部分集合化される様に、単語は定義体によって集合化
される。
そして、集合化された意味要素が其の有意性で連鎖された様に、定義体上の単語
は生命作用で連鎖され、要件事象を成立させている。
例えば、画面は其の画面上の単語を用いて生命体に連鎖を成立させる。
帳表は其の帳表上の単語を用いて生命体に連鎖を成立させる。
ファイルはファイル上の単語を用いて生命体に連鎖を成立させる。
生命体に連鎖を成立させる事に於いて、この定義体は要件事象となり得ている。

ソフトを其の生命作用を代替させる手段として位置づければ、ソフトの役割は定
義体に存在する単語だけでは成立させられない単語の連鎖を成立させる為に新たな
単語を補完して連鎖を成立させるものである。そして、従来のソフトは其の新
たな単語と連鎖の決定は事前にS Eに決めさせると言うものであった。しかし、
本開発方法論で誘導するソフトはこの事を自ら行うと言うものである。
付言すれば、ソフトとは本質的にこの様なものでなければならないのである。

既述した様に、基底論理は集合化された単語は意味領域に転位され、其の意味領域の律性でプログラムに置き換えられたものである。即ち、基底論理は定義体、単語、転位された意味領域で規約されるプログラムの事である。

そして、其の役割は転位された単語の意味性を其の転位された意味領域の律性で置換して其の単語番地にセットする事である。

この単語番地にセットされる意味性は

機能を創り出す意味領域の連鎖を成立させる為に

二つの方法で決定される。ひとつは他の意味領域に存在する同じ単語の意味性を位相的してセットする方法である。もうひとつは同じ意味領域の他の用語の意味性から演算的に導出してセットする方法である。この作用はいずれか一方だけで満たされればよい。そして、両者が共に成立する場合には前者が優先利用される。

前者を位相要素、後者のプログラムを論理要素と呼ぶ。

定義体の全ての単語を意味領域に転位し、其れを基底論理に置き換えた場合、其の意味領域をパレットと呼ぶ。そして、パレットは後述するパレット関数で基底論理を集合化する。

パレット上の基底論理は実行に備えて、例えば、C A L L 命令で順序列化されて並べられる。この並べ方は位相要素と論理要素との群間の順序性は必要となるが、群内での順序性は不要である。

付言すれば、C A L L と基底論理の対は

機械語の命令部とオペランド部の対と機能構造的に等価となる事である。其れは基底論理がオペランド機能と等価である事を意味している。

基底論理はこの意味から従来の機能を表すプログラムではなく、意味領域に存在する単語番地を制御する為の論理となる。これはプログラムは論理で創れると言うひとつの論拠なのである。

単語の意味性を転位した意味領域の律性で置換する仕様は其の意味領域の律性そのものである従来の様に要件機能が仕様でなくてもよい由である。
プログラム化の為の仕様条件である意味領域の律性について述べる。

(1) 受容律性 (W 0 2)

生命作用が行う認識作用に対応して、
意識連鎖を創出する為に逆連想させる処理経路を決定する律性である。
功罪は論じないが、最近のOSではこの意味領域の位相要素は不要である。
論理要素は処理経路の決定を行う。

認識作用とファイルが直接関係する場合、
W 0 3で言う記憶作用がW 0 3に代わって行われる。
(図28参照)

(2) 論理律性 (W 0 3)

単語の意味性を論理化する為の律性である。
論理化の成立とその意識連鎖の帰結とは同義である。
帰結した意識連鎖は不可避的に記憶される。
この場合の記憶とは其の意識連鎖をファイル化する事である。
因に、帳表出力はこの記憶と同義である。
記憶作用は従来型のプログラムで処理してもよいが、
其の機能決定はソフトの立場から言えば不本意な作業となるので、
本開発方法論はこの課題にも対処している。

位相要素は論理化に必要な情報をW 0 2の意味領域から取得する事である。
W 0 2に其の情報が不在の場合、この作用は行う事は出来ない。

論理要素は自己の単語番地に情報が存在しない場合、同じ W 0 3 に存在する自己以外の単語番地の情報を用いて導出する。

導出を成立させる原理や必要な単語番地の情報が存在しない場合、この作用は行う事は出来ない。

(図 2 9 参照)

(3) 相対律性 (W 0 4)

連鎖はさらに連鎖して新たな様相を現す。この意味領域は意識連鎖を更に連鎖する為の律性を司る。

意味領域が画面情報を反映する場合、この意味領域は出力する画面編集を行う。ファイル情報の場合、出力するファイル編集を行う。

位相要素は編集に必要な情報を W 0 2 、 W 0 3 の意味領域から取得する。情報が不在の場合、其の作用を行う事が出来ない。両方の意味領域に存在する場合、 W 0 2 の情報を優先する。

論理要素は編集に必要な情報を同じ W 0 4 の自己以外の単語番地の情報から導出する。

導出を成立させる原理や必要な単語番地の情報が存在しない場合、この作用は行う事は出来ない。

(図 3 0 参照)

(4) パレット (図 3 1 参照)

意識空間に存在する単語は定義体ごとに集合化される。同じ様に、基底論理もこの定義体ごとに集合化される。パレットとはこの基底論理の集合の事である。即ち、パレットは定義体と 3 種の意味領域で区別化される基底論理の集合体の事である。パレットはパレット関数で基底論理を構造化する。

パレット関数の構造は唯ひとつであるがパレットごとに必要である。

(5) W T 単位

従来の要件事象は定義体で規約される。

機能的にまとめられ定義体をW T 単位 (WALK-THROUGH-Unit) と呼ぶ。

(6) パレット連鎖関数 (図 3 2 、図 3 3 参照)

パレット連鎖関数は空間モデルを律する為のもので、意識連鎖を導出する為に意味領域に順序性を与える為の論理である。

パレット連鎖関数には Φ_0 、 Φ_1 で表される二つの種類がある。

前者は基底定義体を画面とするもので、後者はファイルを基底定義体とするものである。尚、後者は前者の部分空間として成立するものである。

(7) 処理経路図

W T 単位で集合化された定義体は処理経路図に書き換えられる。

処理経路図とパレット連鎖関数は等価で、表現の仕方が異なるだけである。即ち、パレット連鎖関数はプログラムであり、処理経路図は設計図面と言うべき仕様書となる。処理経路図は作業着手の初期段階で決定する事が出来る定型化された唯一的な画面である。

処理経路図は従来の要件事象、基本設計書、詳細設計書、プログラム仕様書、テキスト仕様書、操作説明書、開発管理資料、保守管理資料など全ドキュメントの総体の 40 % をカバーする事が出来る。

付則

図23は時間摂理の解説図である。

同じ空間で成立する単位時間のひろがり（解法の総和時間）は其の空間の深さを表す。

図24は空間モデルとその重複構造の解説図である。

補足

本図は基底定義体を画面とするパレット連鎖関数 Φ_0 のパレットに現れる従来型プログラム迄もLYEE化する場合の空間モデルの構造を表す。即ち、従来型プログラム（複数）は基底定義体をファイルとするパレット連鎖関数 Φ_0 （複数）によって T_1 に置換され、其のプログラムが出現した Φ_0 のパレットで律される構造となる。

01. パレット連鎖関数 Φ_0 を処理経路図とするシナリオ関数

$$T_0 = \Phi_0(\{\Phi_{p,k}(L_i, 2, T_1, f)\} + \{\Phi_{p,k}(L_i, 3, T_1, g)\} + \{\Phi_{p,k}(L_i, 4, T_1, q)\})$$

02. パレット連鎖関数 Φ_1 を処理経路図とする従来型プログラムをLYEE化する場合のシナリオ関数

$$T_1 = \Phi_1(\Phi_{p,f}(L_j, 2) + \Phi_{p,f}(L_j, 3) + \Phi_{p,f}(L_j, 4))$$

03. kは画面識別子、f, g, qはファイル識別子である。

04. 処理経路図 Φ_0 は画面を基底とする5種の処理経路で表される。

05. 処理経路図 Φ_1 はファイルを基底とする3種の処理経路で表される。

図25は単語と空間モデルの関係の解説図である。

補足

空間モデルに存在する単語は基底定義体で律されて意味領域に転移される。W02, W03に転位される単語は基底定義体と共に其處に存在する単語が転位される。W03に転位される単語は基底定義体とは関係なく、空間モデルに存在する全単語が転位される。

基底定義体とはパレット連鎖関数が Φ_0 の場合には画面、 Φ_1 の場合にはファイルである。

表中、画面単語とは画面に存在する単語、ファイル単語とはファイルに存在する単語、帳表単語とは帳表に存在する単語の事である。

図26は論理構造の解説図である。

補足

本図はパレットの領域概念とそれに共棲する論理、即ち、基底論理、従来型プログラム P の関係を示す為のものである。尚、従来型プログラムはパレット連鎖関数 Φ_1 で L Y E E 化されるが本図では表現していない。

- 0 1. 基底論理が用いる参照系の F I L E S は省略してある。
- 0 2. 実線 B O X 、太点線 B O X は基底論理の論理を構成するオペランド領域を示す。
- 0 3. 太点線 B O X は従来プログラムの為のデーター処理エリア (B U F F E R) を示す。
- 0 4. 従来プログラム P とそのデーター処理エリアは定義体（画面、帳表、ファイル）の数だけ定義される。
- 0 5. パレットの最大総数は画面数 (G) の $3 \times G$ である。
- 0 6. 実線 B O X の数はシステムに存在する単語 (α) の数 (W) の $3 \times W$ が定義される。
- 0 7. 基底論理の数は定義される空間モデルに存在する単語に依存する実線 B O X の数である。
- 0 8. パレットは基底論理、従来プログラムを要素としてパレット関数 (Φ_p) で統括される。
- 0 9. パレットはパレット連鎖関数 (Φ_0) で支配される。
- 1 0. パレット連鎖関数 (Φ_0) で誘導される論理をシナリオ関数 (T) と呼ぶ。
- 1 1. 既に実行環境として組込済の P K G ソフトを用いる場合の境界条件は従来プログラムで規約する。
- 1 2. 細点線 B O X はパレット連鎖関数が使用する経路指定情報のエリアである。

13. 太細点線 B O X は W F L と呼ばれる。

図27は基底論理の意味の解説図である。

01. バレットはバレット関数で律される基底論理の集合体で、其れ自体がプログラムである。
02. バレットに並べられる同族の基底論理は其の論理構造の性質から順不同でよい。
03. C A L L と基底論理の関係は構造的に C A L L が機械語の命令部、基底論理がオペランドに該当する。
04. 機械語命令はオペランドに作用し、オペランドとは其の命令に必要なデーターエリア番地を指す情報の事である。
05. この関係から、基底論理は①命令処理に必要なデーターエリアの番地を決定し、②其の番地情報から決まる C A L L に代わる演算を行い、③其の結果を目的番地にセットする内容となる。
06. 命令処理に必要なデーターエリア番地は意味領域に存在する転位された単語情報から自ずと判明する。
07. C A L L に代替する演算は意味領域に課せられた普遍的な律性（基底論理のパラダイム）と単語の意味性（演算性）から自ずと決まるものである。
08. 基底論理が其の内部で用いる従来型のプログラムは其れ自体で固有のデーターエリアを定義させなければならない。
09. 基底論理が従来型のプログラムでも C A L L : 基底論理の関係でバレットに並べるのであれば基底論理と同じ様に意味領域のデーターエリアを使用してよい。

備考：理論用語との関係

01. シナリオ関数 (T) は意識連鎖と同義である。
02. バレットは意味領域に存在する単語を基底論理に置き換えた場合の意味領域の事である。
03. 意識連鎖は意識空間で成立する要件事象の存在由来、即ち、真相

を表す論理の事である。

0 4. 意識連鎖の構造は恒常的に唯ひとつとなる。換言すれば、ソフトの構造パラダイムも唯一である。

0 5. 意識連鎖が相対連鎖を経て転位連鎖を連想し、其の転位連鎖の重複が認識空間に存在する森羅万象である。

0 6. 森羅万象とは実相であり状態連鎖である。

0 7. 要件事象とは部分的な森羅万象である。

0 8. 転位連鎖とは成立由来を表す連鎖の事である。

0 9. 転位連鎖の重複とは転位連鎖の連鎖要素にデーター値を与える頻度の事である。

1 0. 転位連鎖の重複はシナリオ関数の重複 (T^n) の事である。

図 28 は基底論理の例 (W 0 2) の解説図である。

単語：売上（画面に存在する項目）

図 29 は基底論理の例 (W 0 3) の解説図である。

単語：売上（画面に存在する項目）

図 30 は基底論理の例 (W 0 4) の解説図である。

単語：売上（画面に存在する項目）

図 31 はパレット関数 (Φ_p) の解説図である。

補足

パレット関数は基底論理、即ち、位相要素と論理要素を集合化してパレットを構成する為の論理である。

0 1. パレット関数はパレット連鎖関数から起動され且つ戻る。

0 2. ファイルの開示閉鎖は実行機器環境との関係から基底論理に振り分けててもよい。

0 3. 最も簡単なパレット関数は C A L L 文の列である。

0 4. パレット再起動フラッグはパレット関数固有のエリアである。

図 32 はパレット連鎖関数 (Φ_0 、基底定義体 = 画面) の解説図である。

補足

0 1. 連鎖の概念図

0 2. パレット連鎖関数と処理経路図は同質である。

0 3. フローの空欄ボックスは無視すること。

図 3 はパレット連鎖関数 (Φ_1 , 基底定義体 = ファイル) 解説図である。

補足

0 1. パレット連鎖関数と処理経路図は同質である。

0 2. フローの空欄ボックスは無視すること。

図面の簡単な説明

図 1 は本発明による作業工程を示す図である。

図 2 は売上を入力するための画面を示す図である。

図 3 は得意先コードを参照するための画面を示す図である。

図 4 は商品入荷の確定を入力するための画面を示す図である。

図 5 は倉庫コードを参照するための画面を示す図である。

図 6 は出荷依頼一覧表を出力操作するための画面を示す図である。

図 7 は出荷依頼一覧表の帳票を示す図である。

図 8 は出荷依頼一覧表の帳票を示す図である。

図 9 は「売上入力」に対する処理経路図を示す図である。

図 10 は「入荷確定入力」に対する処理経路図を示す図である。

図 11 は「出荷一覧表」に対する処理経路図を示す図である。

図 12 は W 0 4 移送要素の一例を示す図である。

図 13 は W 0 4 論理要素の一例を示す図である。

図 14 は W 0 2 論理要素の一例を示す図である。

図 15 は W 0 2 論理要素の一例を示す図である。

図 16 は W 0 3 移送要素の一例を示す図である。

図 17 は W 0 3 論理要素の一例を示す図である。

図 18 はパレット関数の構造パラダイムを示す図である。

図 19 はパレット連鎖関数の構造パラダイムを示す図である。

図20はシナリオ連鎖を説明するための図である。

図21は従来のソフトウェアの作成方法を示す図である。

図22に本発明に係るプログラムの概念的構造を示す図である。

図23は本発明に係る時間撰理を説明するための図である。

図24は本発明に係る空間モデルとその重複構造を説明するための図である。

図25は本発明に係る単語と空間モデルとの関係を説明するための図である。

図26は本発明に係る論理構造を説明するための図である。

図27は本発明に係る基底論理の意味を説明するための図である。

図28は本発明に係る基底論理の例(W02)を示す図である。

図29は本発明に係る基底論理の例(W03)を示す図である。

図30は本発明に係る基底論理の例(W04)を示す図である。

図31は本発明に係るパレット関数を説明するための図である。

図32は本発明に係るパレット連鎖関数(Φ_0)を説明するための図である。

図33は本発明に係るパレット連鎖関数(Φ_1)を説明するための図である。

発明を実施するための最良の形態

即ち、本発明では、まず生産するソフトウェアに必要な画面である定義体識別子を決定する。次に、定義体識別子に存在する単語を抜き出すと共に、定義体識別子に基づきソフトウェアに必要な全てのパレットを処理の流れに沿って配置した処理経路図を作成する。次に、抜き出した単語及び作成した処理経路図に基づき必要なファイルを決定する。次に、抜き出した全ての単語に対し、画面編集またはファイル編集を行う第1の基底論理、処理経路を決定する第2の基底論理及びファイル更新を行う第3の基底論理を作成する。次に、画面単位で前記第1～第3の基底論理をそれぞれ括ってなる3種のパレットを作成すると共に、各パレット内で各基底論理を自律的に有意性をもって実行させるパレット関数を作成する。そして、第1の基底論理に関するパレット関数に基づく画面を送信し、画面を受信して第2の基底論理に関するパレット関数を実行し、この実行結果に基づき第3の基底論理に関するパレット関数の実行を少なくとも1つの処理経路とす

る複数の処理経路から 1 つの処理経路を決定する構造のパレット連鎖関数に、上記の 3 種のパレット関数を組み込む。

以下、本発明をより詳述に説明するために、添付の図面に従ってこれを説明する。

図 1 は本発明による作業工程を示す図である。

即ち本発明では、定義体識別子を決定し（ステップ 101）、単語を決定すると共に（ステップ 102）処理経路図を作成し（ステップ 103）、ファイルを決定し（ステップ 104）、基底論理を作成し（ステップ 105）、パレット関数を作成し（ステップ 106）、パレット連鎖関数へ組み込む（ステップ 107）ことで所望のソフトウェアが作成される。

定義体識別子の決定

例えばある会社における売上を入力するための「売上入力」というソフトウェアを作成する場合を例にとってみると、例えば売上を入力するための画面（図 2）及び得意先コードを参照するための画面（図 3）を決定する。得意先コード参照画面ばかりでなく、必要に応じてこの画面と同様の担当者コード参照画面、取引区分画面、請求先コード画面、商品コード画面等を設定してもよい。

またある会社における商品入荷の確定を入力するための「入荷確定入力」というソフトウェアを作成する場合を例にとってみると、例えば商品入荷の確定を入力するための画面（図 4）及び倉庫コードを参照するための画面（図 5）を決定する。

またある会社における商品出荷依頼の一覧表を出力するための「出荷依頼一覧表」というソフトウェアを作成する場合を例にとってみると、例えば出荷依頼一覧表を出力操作するための画面（図 6）及びその一覧表の帳票（図 7、図 8）を決定する。

単語の決定

単語の決定とは、決定された上記の画面及び帳票から全ての単語を抜き出すことである。

例えば上記の「売上入力」の図 2 の画面では、

「O P C D」
「売上N○」
「売上区分」
「売上日」
「得意先」
「入金日」
「請求先」
「消費税」
「担当者」
「伝單発注N○」
「納品場所」
「TEL」
「備考」
「商品コード」
「数量」
「単価」
「値引」
「金額」
「商品名」
「機種・メディア」
「品番」
「売上合計」
「消費税」
「値引合計」
「総合計」
「実行」
「F1」
「F3」

「F 4」

「F 8」

を抜き出すことができ、これらが決定された単語である。

また上記の「売上入力」の図3の画面では、

「N o」

「得意先」

「得意先名（略称）」

「得意先名」

「住所」

「担当者」

「選択N o」

「得意先名カナ」

「旧コード」

「実行」

「F 12」

を抜き出すことができ、これらが決定された単語である。

また上記の「入荷確定入力」の図4の画面では、

「O P C D」

「倉庫コード」

「H／S区分」

「取次区分」

「実行」

「F 1」

「F 3」

「F 4」

を抜き出すことができ、これらが決定された単語である。

また上記の「入荷確定入力」の図5の画面では、

「N o」

「倉庫コード」

「倉庫名」

「選択No.」

「実行」

「F12」

を抜き出すことができ、これらが決定された単語である。

また上記の「出荷依頼一覧表」の図6の画面では、

「出荷依頼日」「実行」

「F2」

「F3」

を抜き出すことができ、これらが決定された単語である。

また上記の「出荷依頼一覧表」の図7及び図8の帳票では、

「部門」

「納品先」

「請求先」

「住所1」

「住所2」

「TEL」

「部署1」

「担当者」

「店担当」

「出荷依頼日」

「出荷No.」

「商品コード」

「商品名」

「数量」

「更新」

「受注日」

「受注No.」
「行」
「受注担当」
「品番」
「日付」
「貴社注文No.」
「インスト」
「シティ」
「出荷」
「売上」

を抜き出すことができ、これらが決定された単語である。

処理経路図の作成

処理経路図は決定された定義体識別子（画面や帳票）に基づき作成される。処理経路図は人間が作成しても良いし、機械化しても良い。

処理経路図は各画面や各帳票に対応するパレット（ボックスで表示）を線で接続したものである。線はパレット連鎖関数を意味する。画面や帳票が決定されればパレット間の接続は必然的に決まる。

図9は上記「売上入力」に対する処理経路図である。

91はメニュー画面のことであり、メニュー画面から「売上入力」を選択することができる。

92は売上を入力するための画面（図2）に対するW04パレット、93はそのW02パレットである。

94は売上を入力するための画面において「実行」が選択されたときのW03パレットである。

95は売上を入力するための画面において「登録」が選択されたときのW03パレットであり、WFL (Work File area) 96はファイルへの書き込みを意味する。

97は売上を入力するための画面において「得意先コード」の「参照」が選択

されたときの画面（図3）に対するW04パレット、98はそのW02パレットである。

売上を入力するための画面において「終了」が選択されたときはメニュー画面に戻る。

各パレットにおいて、ボックス外上部にはパレットの種別、ボックス内上段には画面の名称、ボックス内下段には画面の識別番号、ボックス外下部の点線ボックスには決定されたファイル名（後述する。）が記述される。

パレットとパレットとを結ぶ線の上部には前の画面で選択されたボタン名（「実行」や「登録」等）、パレットとパレットとを結ぶ線の下部には処理の絶対的種別（R0～R5）が記述される（後述する。）。

図10は上記「入荷確定入力」に対する処理経路図である。

101はメニュー画面のことであり、メニュー画面から「入荷確定入力」を選択することができる。

102は入荷の確定を入力するための画面（図4）に対するW04パレット、103はそのW02パレットである。

104は入荷の確定を入力するための画面において「倉庫コード」の「参照」が選択されたときの倉庫コード参照画面（図5）に対するW04パレット、105はそのW02パレットである。

入荷の確定を入力するための画面において「実行」が選択されたときは入荷の確定を入力するための画面に戻る。

入荷の確定を入力するための画面において「登録」が選択されたときはWFL（Work FiLe aria）106を介してファイルへの書き込みが行われる。

入荷の確定を入力するための画面において「終了」が選択されたときはメニュー画面に戻る。

図11は上記「出荷依頼一覧表」に対する処理経路図である。

111はメニュー画面のことであり、メニュー画面から「出荷依頼一覧表」を選択することができる。

112は出荷依頼一覧表を出力するための画面（図6）に対するW04パレット

ト、113はそのW02パレットである。

出荷依頼一覧表を出力するための画面において「印刷」が選択されたときはWFL (Work File area) 114を介して帳票の出力（印刷）が行われる。

出荷依頼一覧表を出力するための画面において「実行」が選択されたときは出荷依頼一覧表を出力するための画面に戻る。

出荷依頼一覧表を出力するための画面において「終了」が選択されたときはメニュー画面に戻る。

以上のように処理経路図においては、例えはある画面が決まるとその画面での処理内容（実行、登録等の制御単語）から次に必要な処理（画面の表示、帳票の出力、ファイルへの登録、ファイルから読み出し、演算等）が必然的に決まるので、上記処理内容に応じて画面等の間を線で結んでいる。

ファイルの決定

作成された処理経路図及び決定された単語に基づき必要なファイルを決定する。即ち必要なファイルは処理経路図及び単語より必然的に決定できる。決定したファイルは処理経路図のボックス外下部の点線ボックス等に記述する。

例えば図9の符号93で示すW02パレットには、その処理経路図及び単語より必然的に

「従業員M（マスタファイル）」

「名称M」

「顧客情報F（ファイル）」

「部門M」

「W03-S/F」

「商品リストF」

のファイルが必要であることが分かる。

基底論理の作成

各単語に対する基底論理（プログラム）を作成する。即ち、各単語に対する基底論理をデコーディングする。

基底論理には、

W04 移送要素

W04 論理要素

W02 移送要素

W02 論理要素

W03 移送要素

W03 論理要素

の 6 種があり、各単語ごとに 6 種の基底論理（プログラム）を作成する。ただし、
W02 移送要素が不要な場合もある。

例えば「売上」という単語に着目して基底論理を作成する例を説明する。

W04 移送要素は図 12 に示すように作成する。

まず該当する W02 売上の項目（ファイル）に売上値が存在するかどうかを判断する（ステップ 1201）。存在する場合にはその売上値を W04 売上の項目にセットし（ステップ 1202）、処理を終了する。

存在しない場合には W03 売上の項目に売上値が存在するかどうかを判断する（ステップ 1203）。存在する場合にはその売上値を W04 の売上の項目にセットし（ステップ 1204）、処理を終了する。存在しない場合には、そのまま処理を終了する。

単語によっては必要に応じて、ステップ 1201 に前に実行有無を確認するステップを設ける。また、ステップ 1204 の後に目的処理の成立の有無を確認するステップを設け、目的処理が成立していないときにはパレットを再起動するフラグをセットする。

重要なことは如何なる単語に対しても常にこの構造でプログラムを作成する事であり、このことは以下のプログラムについても同じである。

W04 論理要素は図 13 に示すように作成する。

まず W04 売上の項目に売上値が入力済みかどうかを判断する（ステップ 1301）。入力済みの場合には処理を終了する。

入力済みでない場合には、W04 売上の項目に対する売上値が編集可能であるかどうかを判断する（ステップ 1302）。即ち、例えば売上値が価格 × 個数で

あるとすると、W04価格の項目及びW04個数の項目に既に数値が存在するかどうかを判断する。編集可能であれば編集し（例えば存在する価格と個数から売上を求める。）（ステップ1303）、編集結果（例えば求められた売上値）をW04売上の項目にセットする（ステップ1304）。

編集できない場合には、必要に応じて（ステップ1305）パレットを再起動するフラグをセットする（ステップ1306）。即ち、売上値の決定は一旦留保され、上記の例でいえば価格と個数の入力があるまでその決定が留保される。このことは、つまり本発明に係るプログラムが自律的に有意性を決定している事に他ならない。

W02論理要素は図14に示すように作成する。

まず売上値が入力済みか（W04売上の項目にセットされているか）どうかを判断する（ステップ1401）。入力済みでない場合には処理を終了する。

入力済みの場合には入力売上値をW02売上の項目にセットし（ステップ1402）、処理を終了する。

単語によっては必要に応じて、ステップ1401に前に実行有無を確認するステップを設ける。また、ステップ1402の後に目的処理の成立の有無を確認するステップを設け、目的処理が成立していないときにはパレットを再起動するフラグをセットする。

なお、例えば単語「実行キイ」のような制御単語の場合には、上記のステップ1402が処理経路フラグをセットする処理となる（図15参照）。処理経路フラグ（R=0～5、ただしR=0は例外的）は処理経路図で決まる。これにより後述するようにパレット連鎖関数において経路を振り分ける処理が可能となる。

W03論理要素は図16に示すように作成する。

まずW02売上の項目に売上値が入力済みかどうかを判断する（ステップ1601）。入力済みの場合にはその売上値をW03売上の項目にセットし（ステップ1602）、処理を終了する。

存在しない場合にはそのまま処理を終了する。

単語によっては必要に応じて、ステップ1601に前に実行有無を確認するステップを設ける。また、ステップ1602の後に目的処理の成立の有無を確認するステップを設け、目的処理が成立していないときにはパレットを再起動するフラグをセットする。

W03論理要素は図17に示すように作成する。

まずW03売上の項目に売上値が入力済みかどうかを判断する（ステップ1701）。入力済みの場合には処理を終了する。

入力済みでない場合には、W03売上の項目に対する売上値が導出可能であるかどうかを判断する（ステップ1702）。可能でないような場合には処理を終了する。可能な場合には計算可能かどうかを判断する（ステップ1703）。即ち、例えば上記と同様に売上値が価格×個数であるとすると、W03価格の項目及びW03個数の項目に既に数値が存在するかどうかを判断する。そして、導出可能であれば計算し（例えば存在する価格と個数から売上を求める。）（ステップ1704）、計算結果（例えば求められた売上値）をW03売上の項目にセットする（ステップ1705）。

計算できない場合には、必要に応じて（ステップ1706）パレットを再起動するフラグをセットする（ステップ1707）。即ち、この場合も上記同様に売上値の決定は一旦留保され、上記の例でいえば価格と個数の入力があるまでその決定が留保される。このことは、つまり本発明に係るプログラムが自律的に有意性を決定している事に他ならない。

パレット関数の作成

W04、W02、W03に対するパレット関数の作成を行う。図18はパレット関数の構造パラダイムを示しており、それぞれについてこのような構造のパレット関数（プログラム）を作成すればよい。

まずファイルをオープンする（ステップ1801）。即ち、例えばW04に関するパレット関数では複数のW04パレットから所定の1つのW4パレットに関するファイルをオープンする。そして、各移送要素と論理要素を順次実行する（ステップ1802、1803）。

その後、パレット起動フラグがセットされている場合には（ステップ1804）、フラグをリセットする（ステップ1805）と共に各移送要素と論理要素を順次再実行する（ステップ1802、1803）。即ち、パレット関数は、各パレットのパレット起動フラグと連動してプログラムの自律的な有意性の決定を可能としている。

一方、パレット起動フラグがセットされていない場合にはファイルを閉じて（ステップ1806）処理を終了する。

パレット連鎖関数への組み込み

パレット連鎖関数の構造はいかなるソフトウェアにおいても同一である。従って、予め作成してあるパレット連鎖関数に上述のように作成した基底論理やパレット関数を組み込めばよい。

パレット連鎖関数の構造パラダイムを図19に示す。

即ち、まず送信情報を決定する（ステップ1901）。送信情報とは、表示すべき画面の事である。例えば図9に示した処理経路図を参照しながら説明すると、例えば売上入力の画面（図2参照）において「得意先コード」の「参照」が選択される（「得意先コード」のW02パレットでR=3の処理経路フラグが立っている。）と得意先コード照会の画面（図3参照）を送信情報と決定する。

送信情報にシステムを閉鎖する情報が含まれているような場合には（ステップ1902）、システムを閉鎖する（ステップ1903）。システムを閉鎖する情報とは、例えば図9に示した処理経路図を参照しながら説明すると、例えば売上入力の画面（図2参照）において「F3」が選択されたこと（「F3」のW02パレットでR=0の処理経路フラグが立っていること。）である。システムを閉鎖するとは、例えば上記の例で言えばメニュー画面に戻る事である。

次に、WT単位内の継続かどうかを判断する（ステップ1904）。これは例外的処理に関するものである。即ち、WT単位とは、図20に示すように、例えばデータの登録に使用される登録画面と当該データの登録の際のデータの参照に使用される参照画面とを1つの単位で括ってなるものである。例えば図9に示した処理経路図を参照しながら説明すると、売上入力の画面（図2参照）と得意先

コード照会の画面（図3参照）とを1つの単位として括ってなるものがWT単位である。しかし、他の処理経路図、例えば入荷確定入力（図10、図4参照）の画面が必要となる場合には、これらと連鎖する必要がある。そこで、図20に示すように、必要に応じてシナリオ連鎖、即ち他のWTとの間で連鎖を行う（ステップ1905、1906）。これは巨大なプログラムに対応する場合に特に有効となる。

WT単位内の継続の場合には、当該WT単位の全パレットをセットする（ステップ1907）。例えば図9に示した処理経路図を参照しながら説明すると、この処理全体に係る全てのパレット（W02～W04）をそれぞれワーキングメモリ上にセットする。

そして、まず該当するW04パレットを実行する（ステップ1908）。即ち、図18に示したパレット関数に全てのW04移送要素とW04論理要素がセットされたW04パレットを実行する。これにより表示すべき画面に関するデータが決定される。

そして、送信機能を実行する（ステップ1909）。即ち、データ等がセットされた画面を送信する（表示手段、例えばCRTに送る。）。

その後、受信機能を実行する（ステップ1910）。即ち、データ等が入力された画面を受信する（表示手段、例えばCRTから受ける。）。

そして、受信データが正常かどうかを判断し（ステップ1911）、異常の場合には最初からやり直す。即ち、規則違反のデータ等がないかを判断する。

次に、該当するW02パレットを実行する（ステップ1912）。即ち、図18に示したパレット関数に全てのW02論理要素がセットされたW02パレットを実行する。これにより入力されたデータ等が決定される。

次に、処理経路が決定される（ステップ1913、1914）。処理経路は上述したようにW02論理要素上の処理経路フラグ（R=1～5）によって判断される。

処理経路フラグがR=1の場合には、該当するW03パレット（ファイルへの登録を伴わないW03パレット）を実行する（ステップ1915）。例えば図9

に示した W03 パレット 94 のラインを実行する。そして、最初の処理（ステップ 1901）に戻る。即ち、処理経路 R = 1 はデータ処理の実行のみを行う経路処理である（図 9 参照）。

処理経路フラグが R = 2 の場合には、該当する W03 パレット（ファイルへの登録を伴う W03 パレット）を実行する（ステップ 1916）。例えば図 9 に示した W03 パレット 95、WFL96 のラインを実行する。そして、最初の処理（ステップ 1901）に戻る。処理経路 R = 2 はデータ処理の実行とデータのファイルへの登録を行う経路処理である（図 9 参照）。

処理経路フラグが R = 3 ~ 5 の場合には、そのまま最初の処理（ステップ 1901）に戻る。

ここで、処理経路 R = 3 はそのまま W04 パレット（同一、異種の両方）に戻る経路処理である（図 9 参照）。

経路処理 R = 4 はデータをそのままファイルに登録する処理である（図 10 参照）。

処理経路 R = 5 はデータをそのままファイルから取り出す処理である（図 11 参照）。

以上のことによって本発明に係るソフトウェアは作成される。ここで、従来方法を図 21 に示し、図 1 に示した本発明方法とを比較する。従来方法のステップ 2101 ~ 2103 の工数と本発明方法のステップ 101 ~ 104 の工数とを比較すると本発明方法が約 1/5 の工数となり、従来方法のステップ 2104 ~ 2106 の工数と本発明方法のステップ 105 ~ 107 の工数とを比較すると本発明方法が約 1/10 ~ 1/15 の工数となる。

図 22 に本発明に係るプログラムの概念的構造を示す。

即ち複数の W02 パレット（各パレットは k 個の $l_{i,2}$ 、k 個の $l_{i,2}$ を含む。）、W03 パレット（各パレットは k 個の $l_{i,3}$ 、k 個の $L_{i,3}$ を含む。）及び W04 パレット（各パレットは k 個の $l_{i,4}$ 、k 個の $L_{i,4}$ を含む）をそれぞれパレット関数 (Φ_p) に組み込み、これをパレット連鎖関数 (Φ_0) に組み込ん

だものである。つまり

$$T_0 = \Phi_0 (\{\Phi_p (l_{i,2}, L_{i,2})\}_k + \{\Phi_p (l_{i,3}, L_{i,3})\}_k \\ + \{\Phi_p (l_{i,4}, L_{i,4})\}_k)$$

が成立していることが分かる。

図24は基底定義体を画面とするパレット連鎖関数 Φ_0 のパレットに現われる従来型プログラム迄もLYEE化する場合の空間モデルの構造を表す。即ち、従来型プログラム（複数）の基底定義体をファイルとするパレット連鎖関数 Φ_0 （複数）によって T_1 に置換され、其のプログラムが出現した Φ_0 のパレットで律された構造となる。

図33は従来プログラムを基底論理と同等に取り扱う為のプログラム、即ち従来プログラムをLYEE化する為のパレット連鎖関数 Φ_1 を示すフーチャートである。図33のプログラムは基底論理と同様にパレット関数に組み込まれる。

産業上の利用可能性

これ迄のソフト開発作業では属人的能力への依存度合が極めて高く、且つ、其れが作業者全員に求められる所に根本的な問題があった。本発明は思考方法の深層にある意識作用を認識作用との関係に於いて理論的に捉え、其れを開発方法論化したもので、OS・ミドル・ゲーム・制御・業務など分野を問わず適用する事が出来る。

本発明で誘導されるソフト構造は理論的に決定される事から、再帰性があり、且つ、唯一的となる。其の結果、開発されたソフトウェアはブラックボックスでなくなり、品質的には人為的誤謬が排除され、構造が鮮明でシステムが強靭になる。其れ故、見積もりもより正確に行う事が可能となり、開発計画の策定、開発管理がやり易くなる。

さらに従来に比べ開発期間は1/2~1/4、開発原価は1/3~1/5、開発作業者の生産性は10~15倍、開発ドキュメント量は1/15~1/20と

なるなど驚異的である。さらに保守性は 50 ~ 75 倍に到達する。

従来方法では保守作業の生産性は開発作業の生産性の 10 分の 1 程度とされる。

本発明では開発作業の生産性と等価になるので、開発作業者の生産性の 10 倍の生産性が保守性の生産性と言う事になる。但し、開発作業者の生産性は従来で言う作業工程、即ち、要件定義、基本設計、詳細設計、プログラム作成、検証の全作業域に対する生産比較値であるので、保守作業の作業域概念を其の 2 分の 1 の範囲に適用するとした場合の値である。

請求の範囲

1. 生産するソフトウェアに必要な画面である定義体識別子を決定する工程と、
前記定義体識別子に存在する単語を抜き出す工程と、
前記定義体識別子に基づき前記ソフトウェアに必要な全てのパレットを処理の流れに沿って配置した処理経路図を作成する工程と、
前記抜き出した単語及び前記作成した処理経路図に基づき必要なファイルを決定する工程と、
前記抜き出した全ての単語に対し、画面編集またはファイル編集を行う第1の基底論理、処理経路を決定する第2の基底論理及びファイル更新を行う第3の基底論理を作成する工程と、
画面単位で前記第1～第3の基底論理をそれぞれ括ってなる3種の前記パレットを作成すると共に、各パレット内で各基底論理を自律的に有意性をもって実行させるパレット関数を作成する工程と、
前記第1の基底論理に関するパレット関数に基づく画面を送信し、画面を受信して前記第2の基底論理に関するパレット関数を実行し、この実行結果に基づき、前記第3の基底論理に関するパレット関数の実行を少なくとも1つの処理経路とする複数の処理経路から1つの処理経路を決定する構造のパレット連鎖関数に、上記の3種のパレット関数を組み込む工程と
を具備することを特徴とするソフトウェアの生産方法。
2. 生産するソフトウェアに必要な画面に存在する全ての単語に対して作成された、画面編集またはファイル編集を行う第1の基底論理、処理経路を決定する第2の基底論理及びファイル更新を行う第3の基底論理とを有する第1の手段と、
画面単位で前記第1～第3の基底論理をそれぞれ括ってなる3種の各パレット内で各基底論理をそれぞれ自律的に有意性をもって実行させる第2の手段と、
前記第1の基底論理に関する第2の手段の実行に基づく画面を送信し、画面を受信して前記第2の基底論理に関する第2の手段を実行し、この実行結果に基づき、前記第3の基底論理に関する第2の手段の実行を少なくとも1つの処理経路とする複数の処理経路から1つの処理経路を決定する第3の手段と

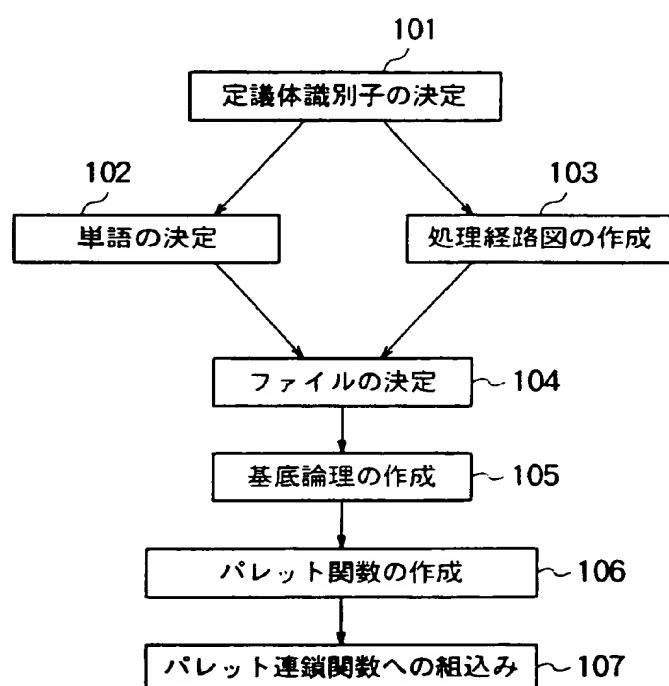
を具備することを特徴とする処理装置。

3. 生産するソフトウェアに必要な画面に存在する全ての単語に対して作成された、画面編集またはファイル編集を行う第1の基底論理、処理経路を決定する第2の基底論理及びファイル更新を行う第3の基底論理とを有する第1の手段と、画面単位で前記第1～第3の基底論理をそれぞれ括ってなる3種の各パレット内で各基底論理をそれぞれ自律的に有意性をもって実行させる第2の手段と、

前記第1の基底論理に関する第2の手段の実行に基づく画面を送信し、画面を受信して前記第2の基底論理に関する第2の手段を実行し、この実行結果に基づき、前記第3の基底論理に関する第2の手段の実行を少なくとも1つの処理経路とする複数の処理経路から1つの処理経路を決定する第3の手段と

を具備することを特徴とする記録媒体。

図 1



2

3

4

5

6

図 7

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

8

図 9

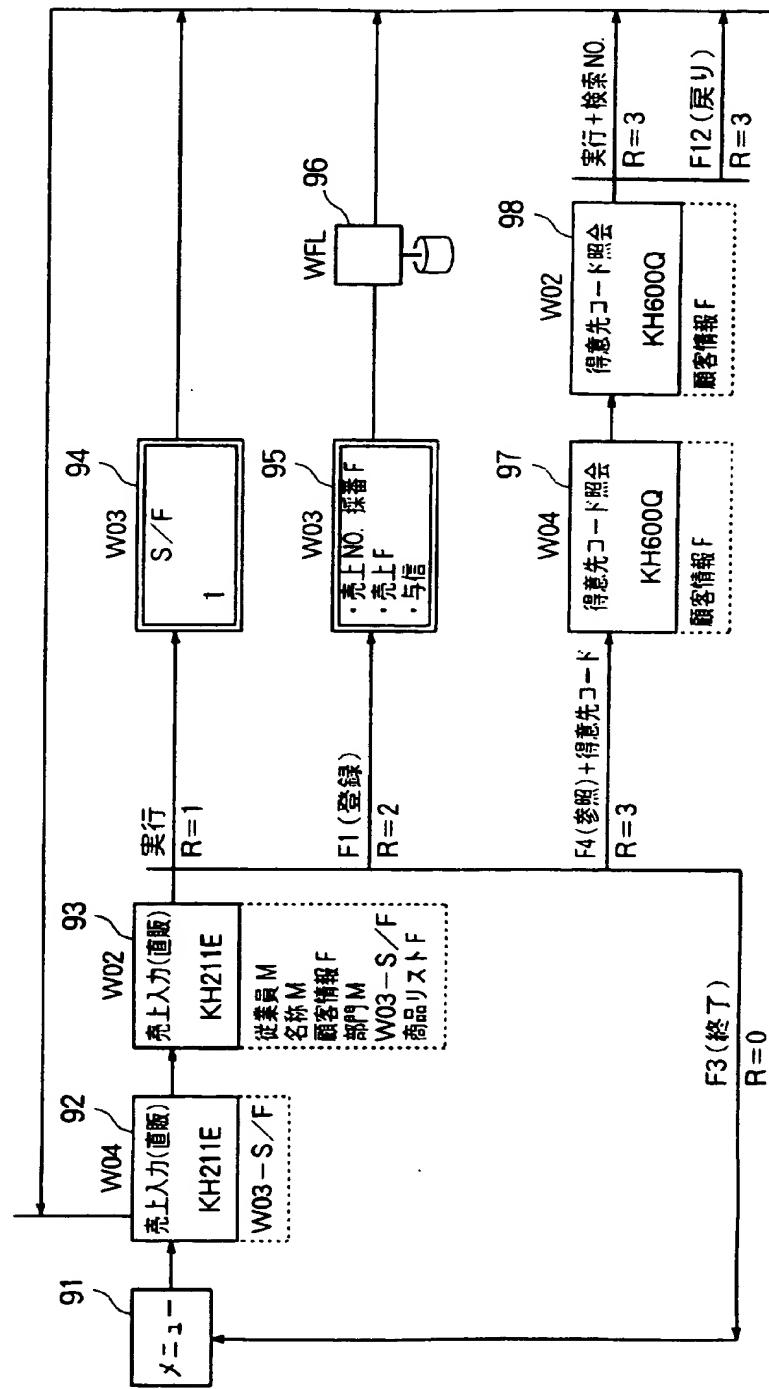


図 10

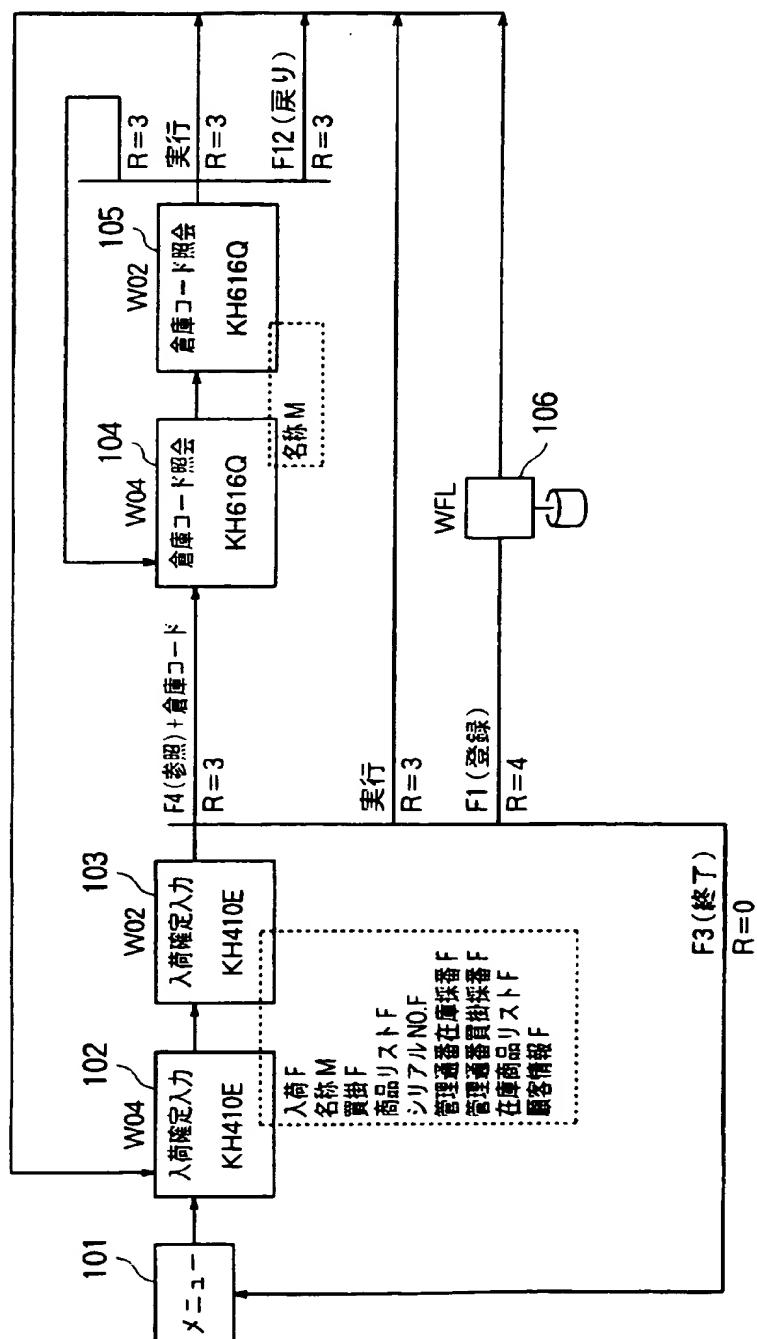


図 11

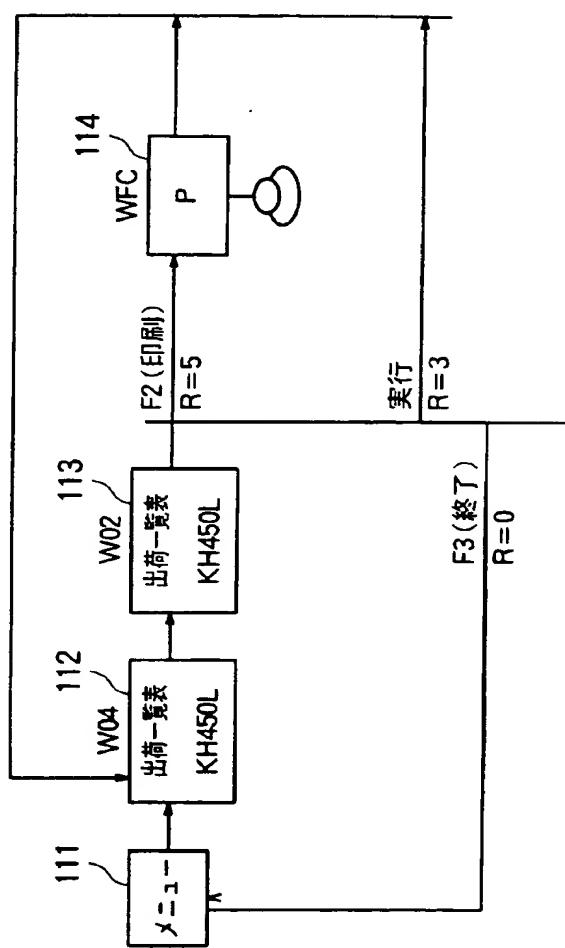


図 12

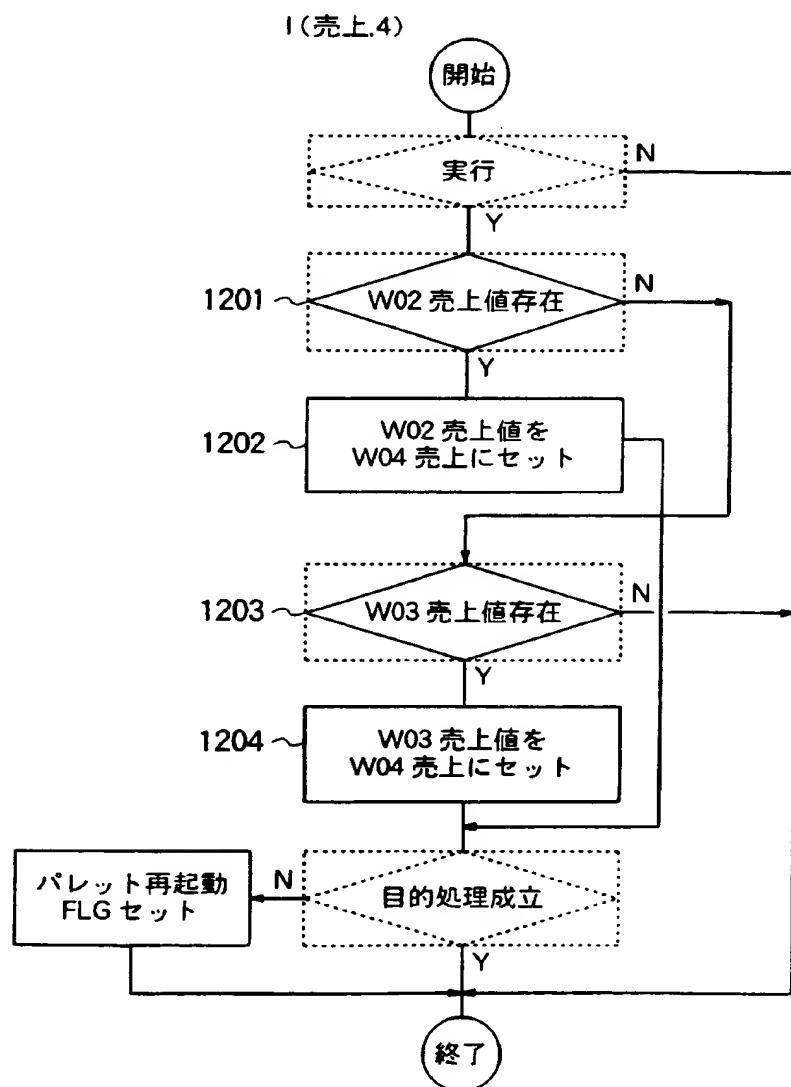


図 13

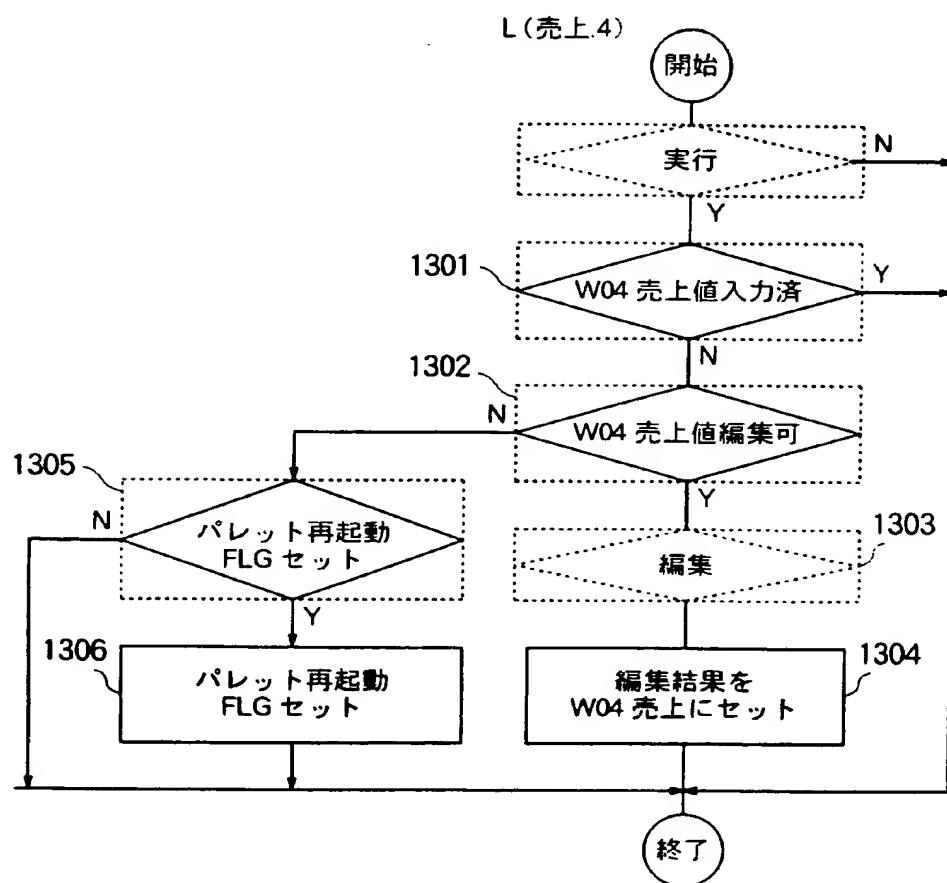


図 14

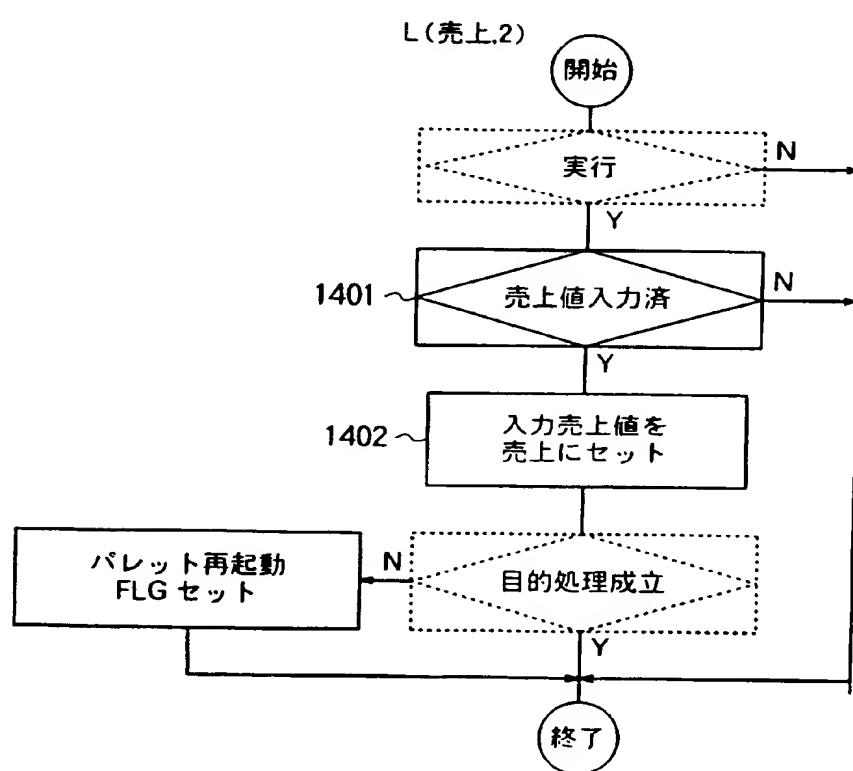


図 15

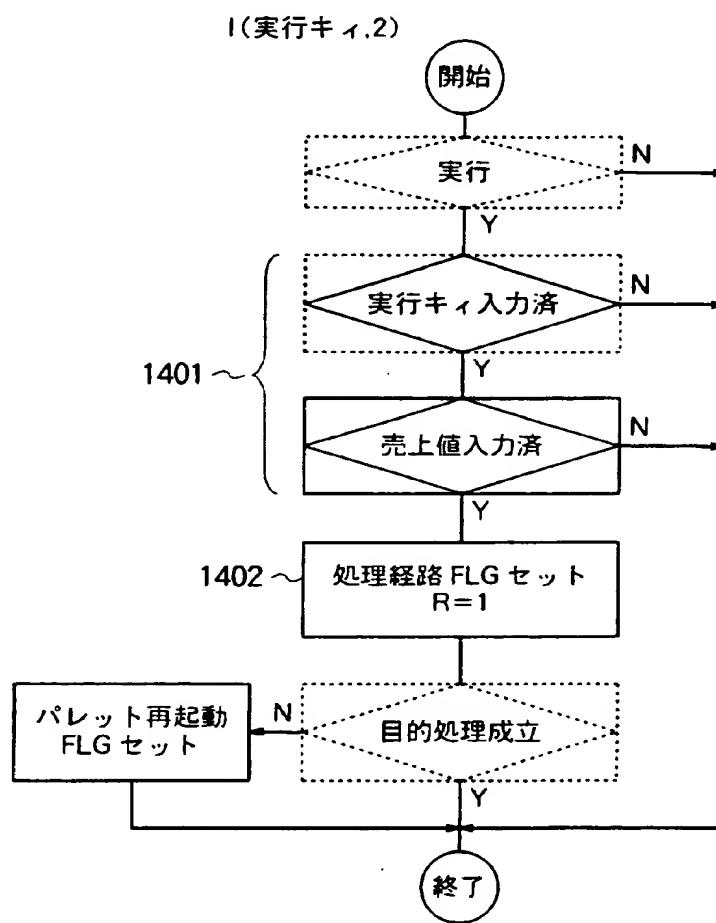


図 16

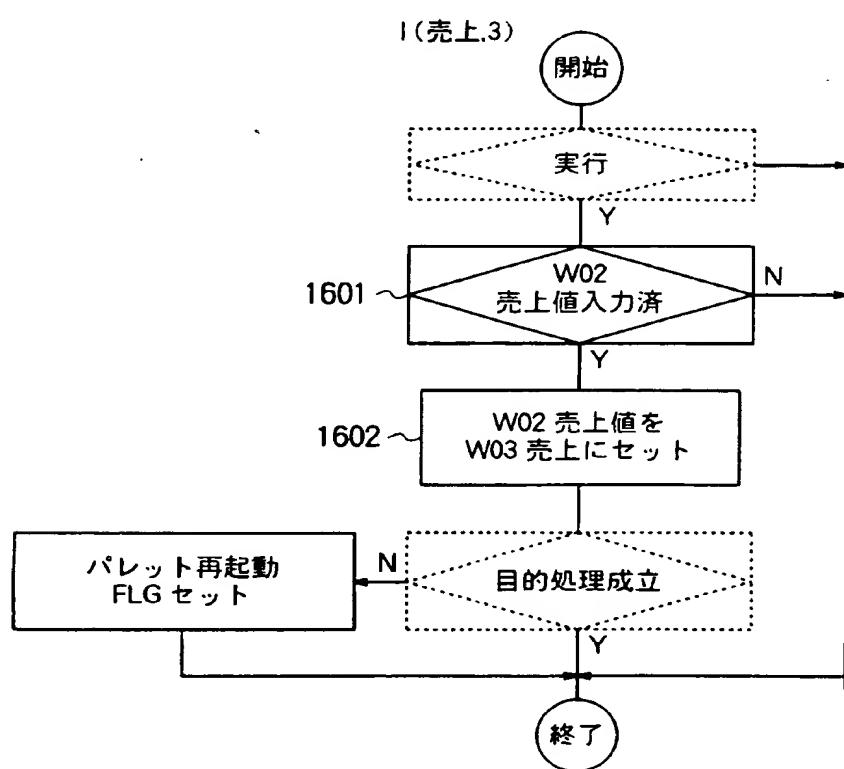


図 17

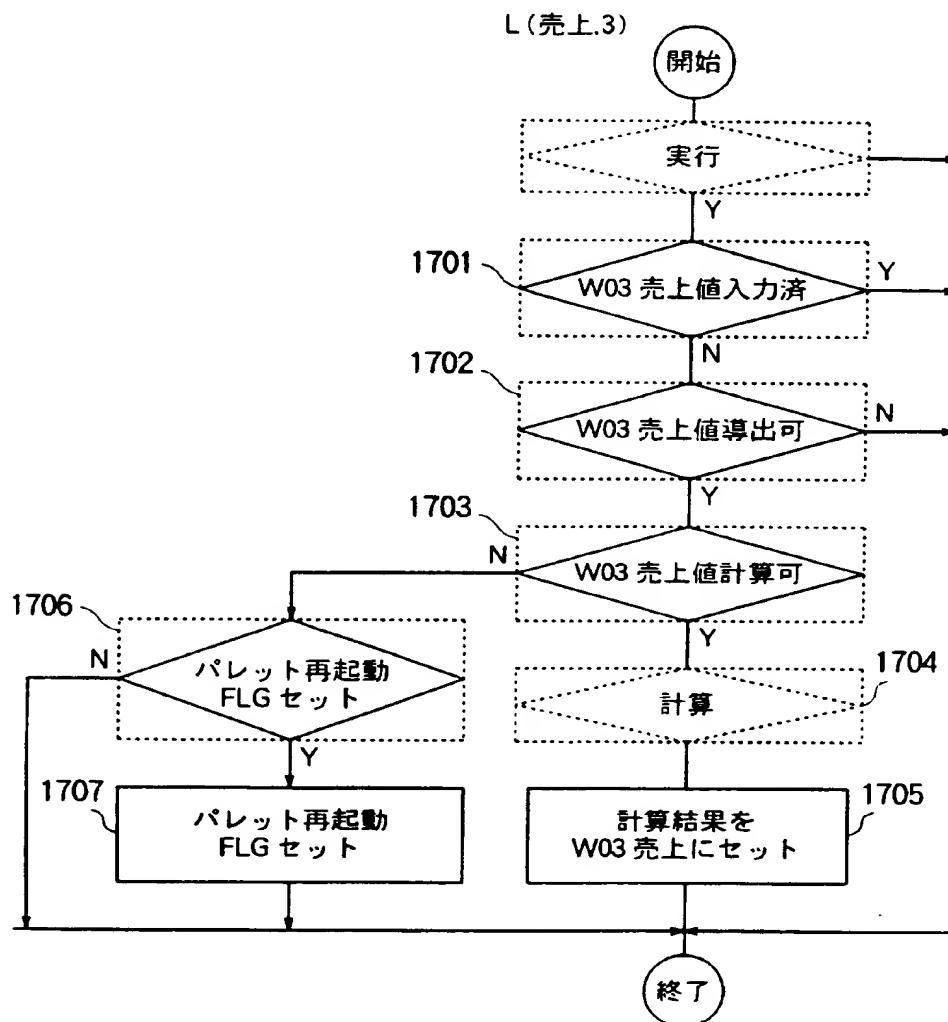


図 18

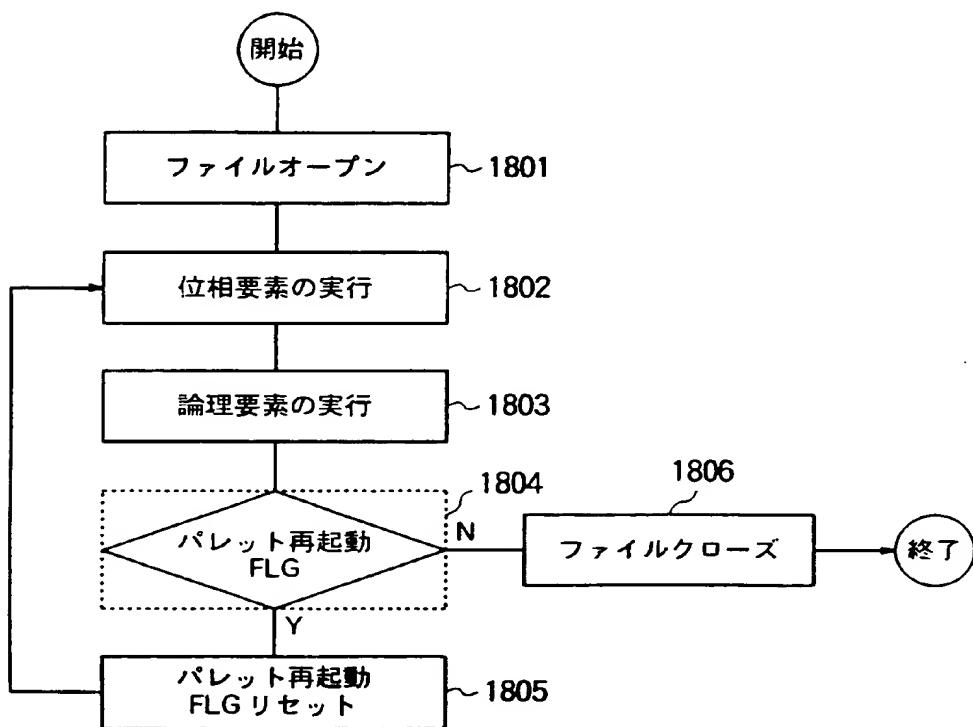


図 19

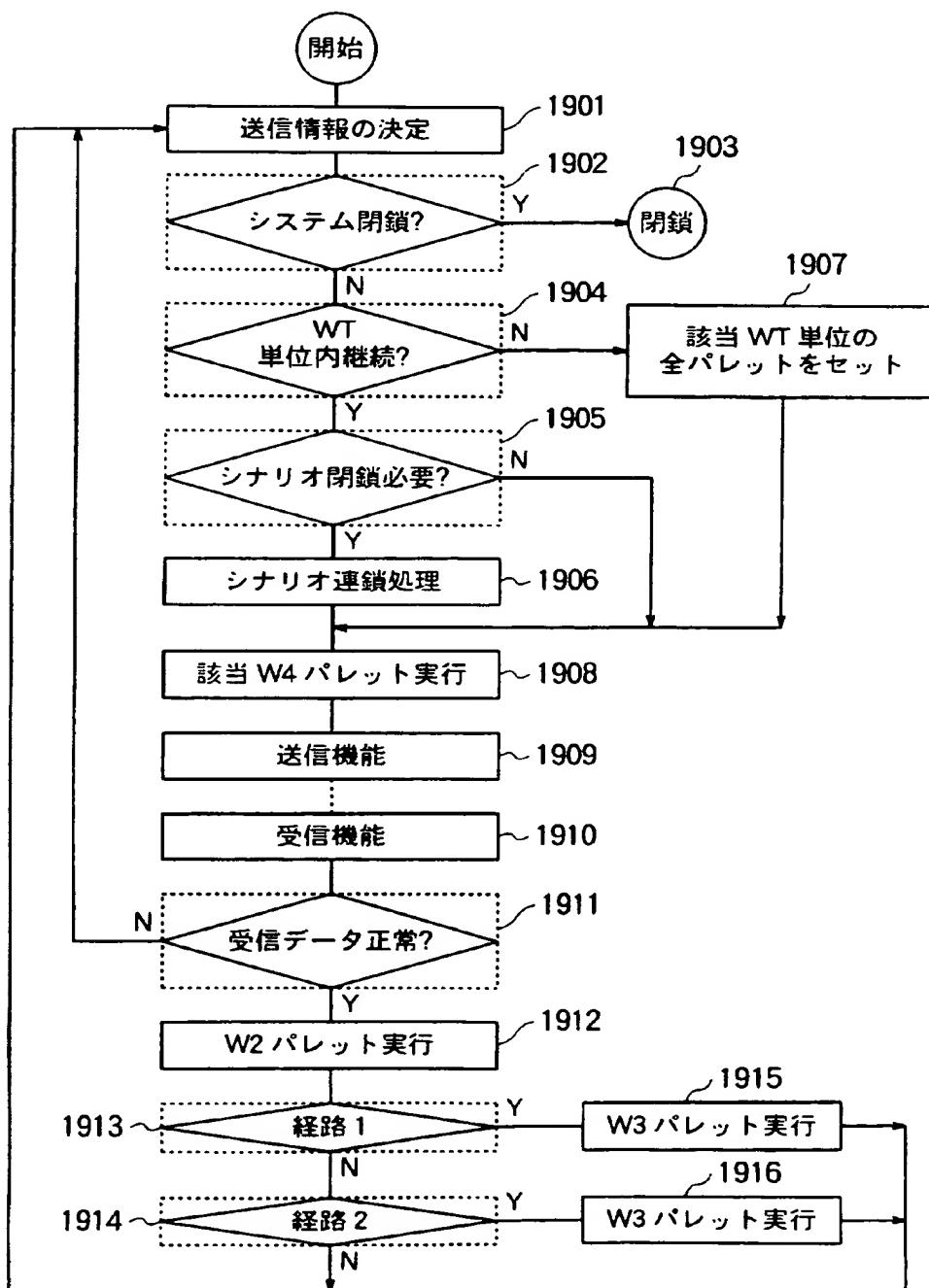


図 20

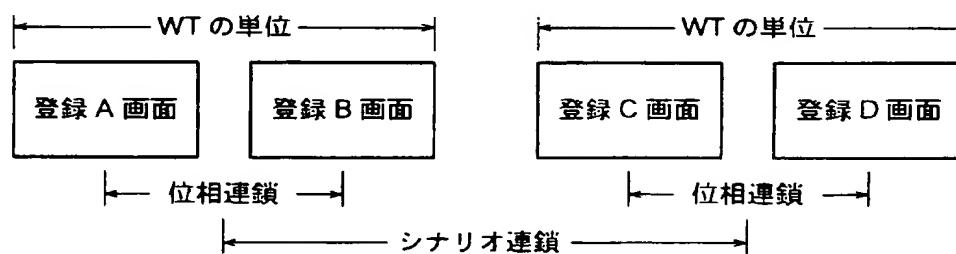


図 21

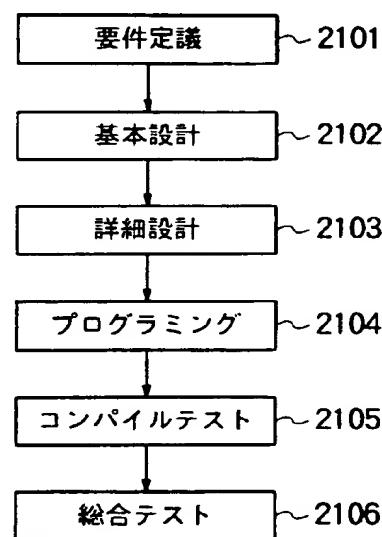


図 22

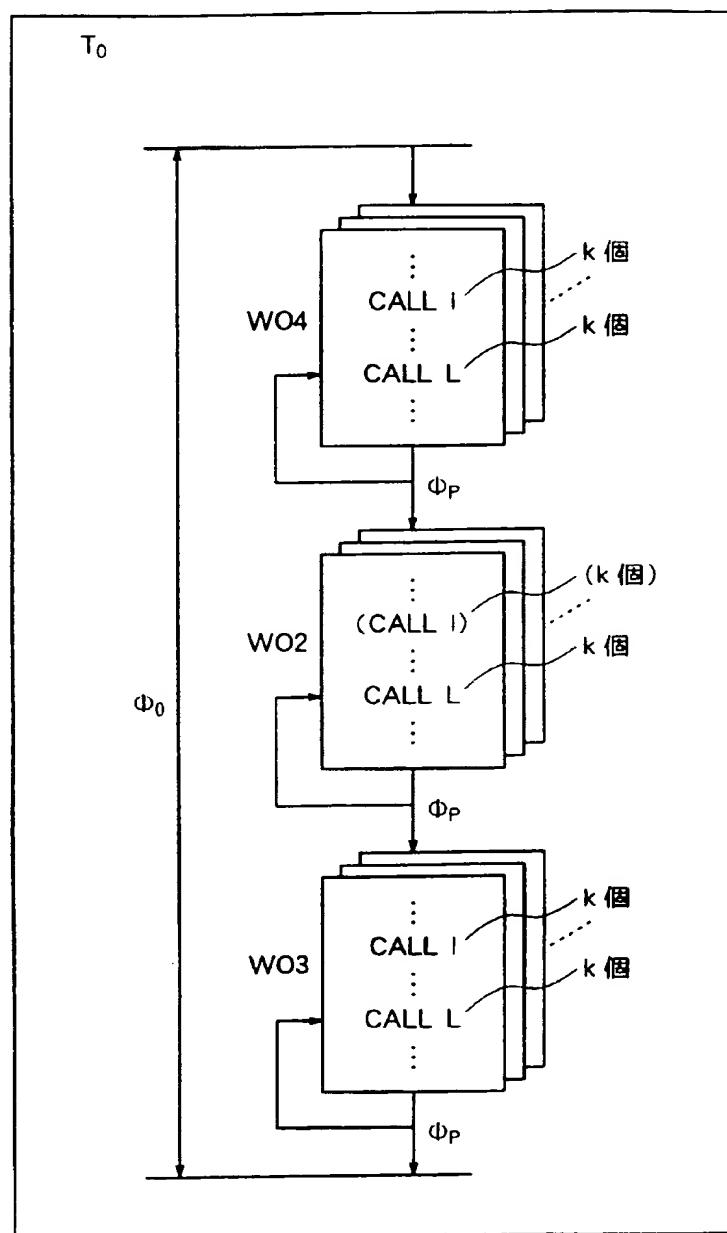


図 23

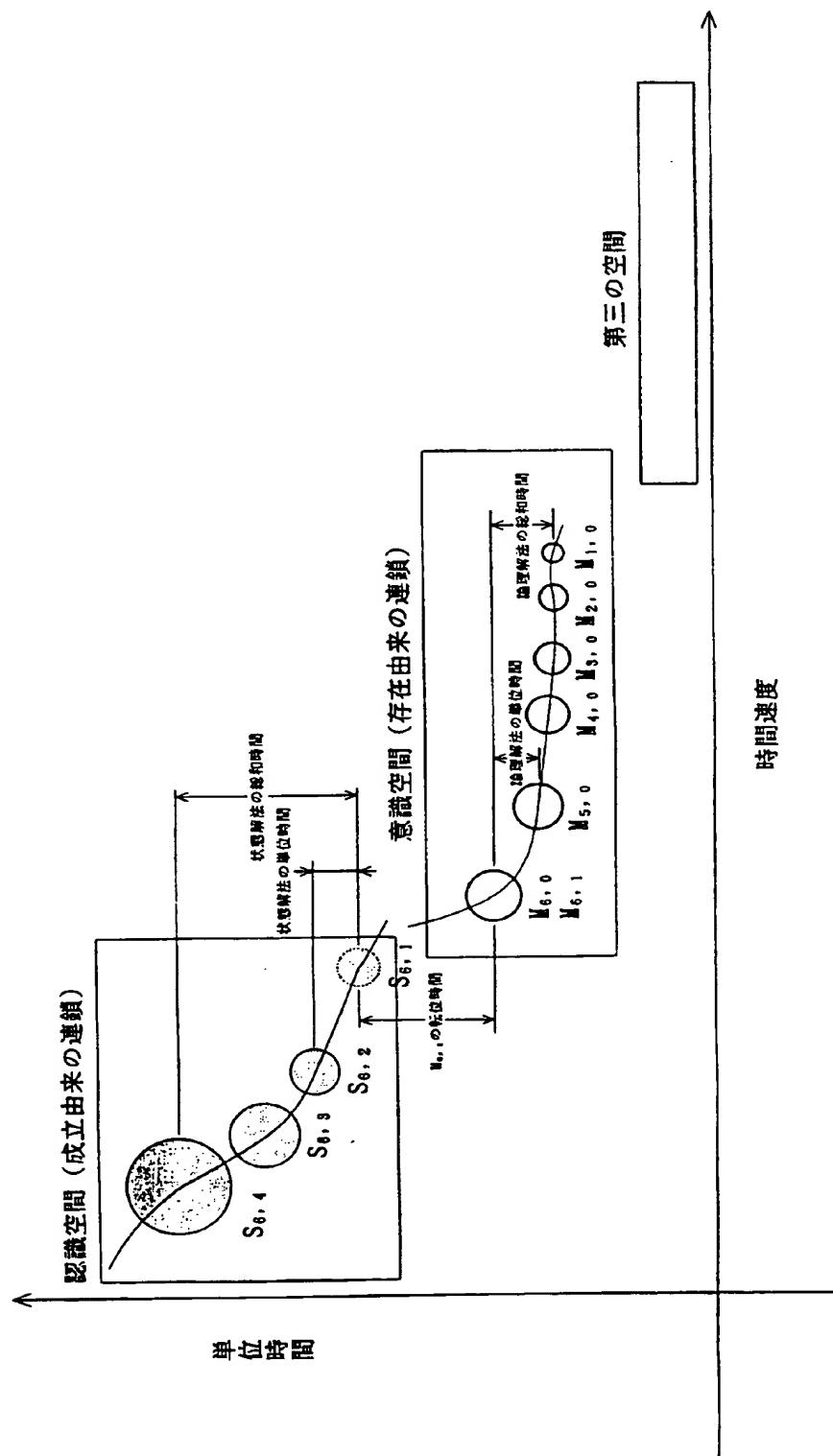


図 24

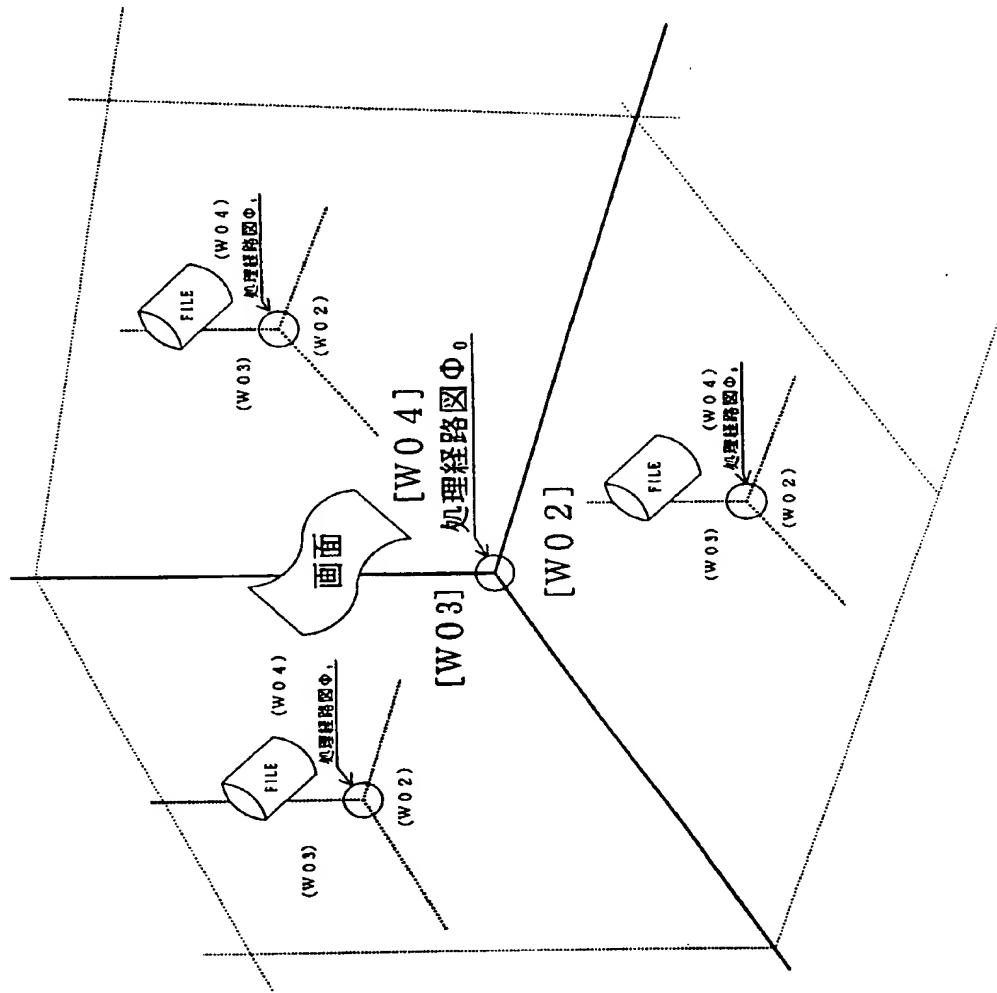


図 25

処理経路図		意味領域					
ルート選択	構成要素	W 0 3			W 0 4		
Φ_0	パレット	W 0 2			W 0 4		
	基底定義体	画面			画面		
	意味要素	画面単語			画面単語		
Φ_1	パレット選択	$T_0 = \phi_0 ([\phi_{0,1}, [\mathcal{L}_{0,1}, T_{0,1}]), [\phi_{0,2}, [\mathcal{L}_{0,2}, T_{0,2}]), [\phi_{0,3}, [\mathcal{L}_{0,3}, T_{0,3}]), [\phi_{0,4}, [\mathcal{L}_{0,4}, T_{0,4}])$					
	パレット	W 0 2	W 0 3	W 0 2	W 0 3	W 0 2	W 0 4
	基底定義体	FILE	無	FILE/帳表	無	FILE/帳表	無
	意味要素	FILE単語	全単語	FILE/帳表	全単語	FILE/帳表	全単語
	パレット選択	$T_{1,1} = \phi_1 ([\phi_{1,1}, [\mathcal{L}_{1,1}, \phi_{1,2}], [\mathcal{L}_{1,2}, \phi_{1,3}], [\mathcal{L}_{1,3}, \phi_{1,4}])$	$T_{1,2} = \phi_1 ([\phi_{1,1}, [\mathcal{L}_{1,1}, \phi_{1,2}], [\mathcal{L}_{1,2}, \phi_{1,3}], [\mathcal{L}_{1,3}, \phi_{1,4}])$	$T_{1,3} = \phi_1 ([\phi_{1,1}, [\mathcal{L}_{1,1}, \phi_{1,2}], [\mathcal{L}_{1,2}, \phi_{1,3}], [\mathcal{L}_{1,3}, \phi_{1,4}])$	$T_{1,4} = \phi_1 ([\phi_{1,1}, [\mathcal{L}_{1,1}, \phi_{1,2}], [\mathcal{L}_{1,2}, \phi_{1,3}], [\mathcal{L}_{1,3}, \phi_{1,4}])$	$T_{1,5} = \phi_1 ([\phi_{1,1}, [\mathcal{L}_{1,1}, \phi_{1,2}], [\mathcal{L}_{1,2}, \phi_{1,3}], [\mathcal{L}_{1,3}, \phi_{1,4}])$	$T_{1,6} = \phi_1 ([\phi_{1,1}, [\mathcal{L}_{1,1}, \phi_{1,2}], [\mathcal{L}_{1,2}, \phi_{1,3}], [\mathcal{L}_{1,3}, \phi_{1,4}])$

図 26

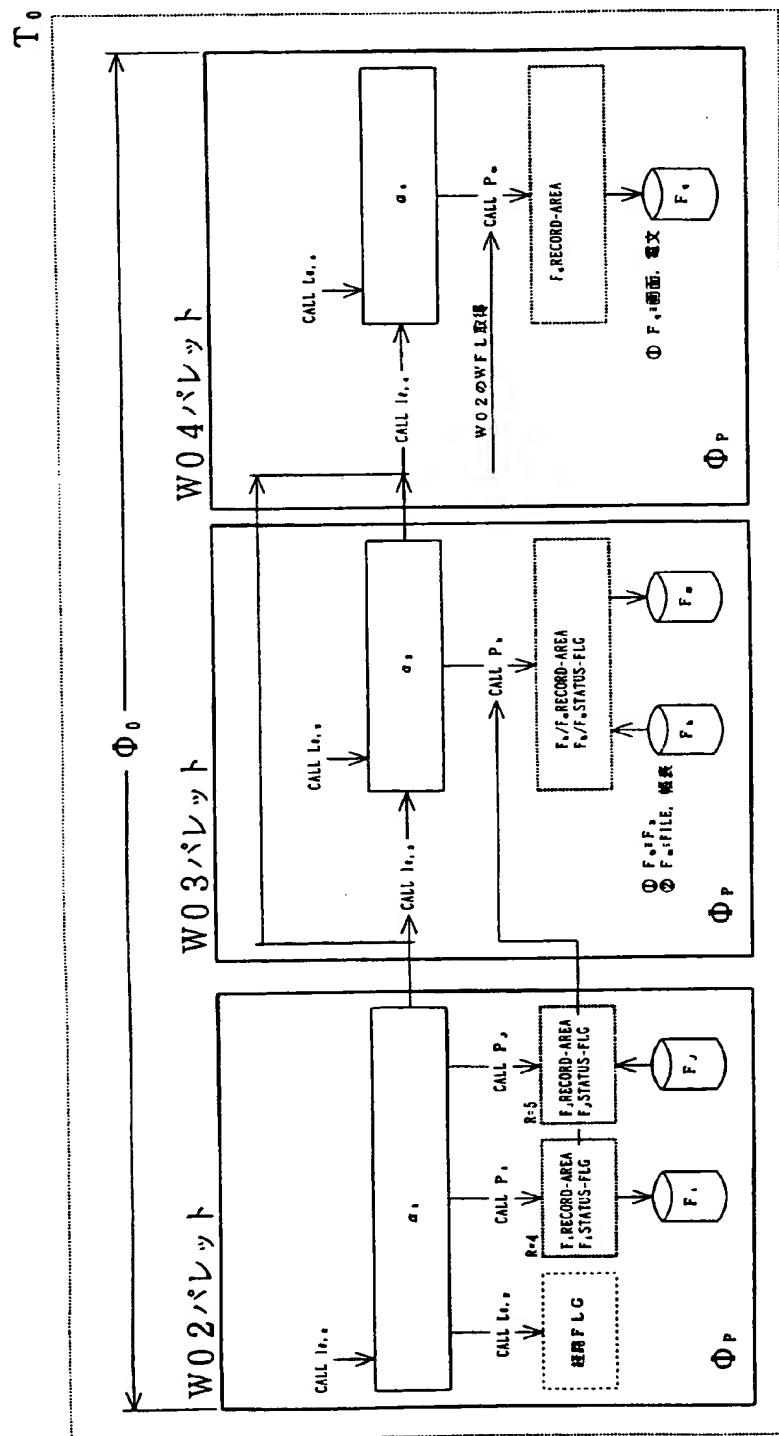
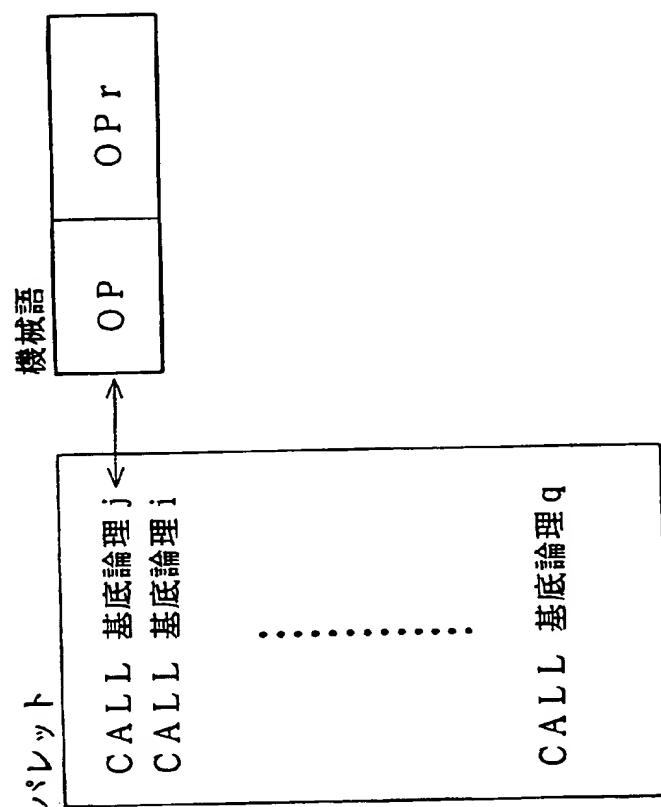
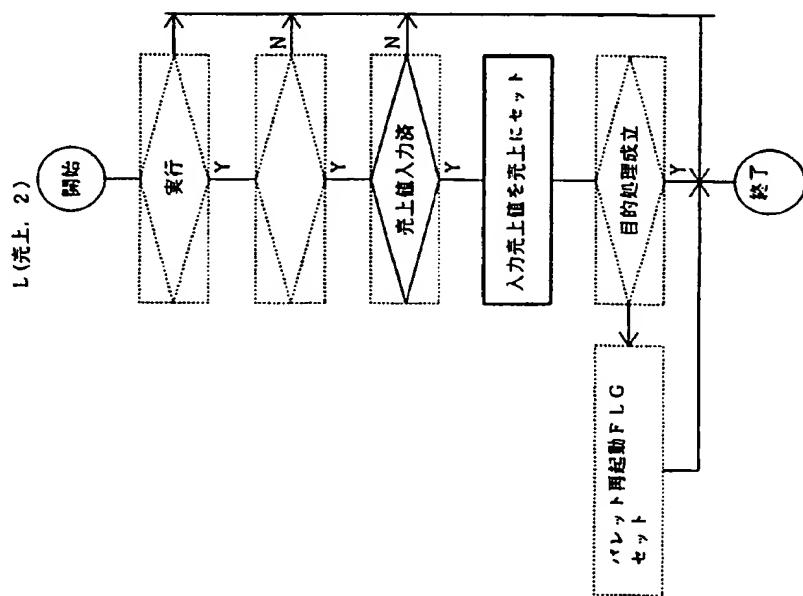


図 27





28

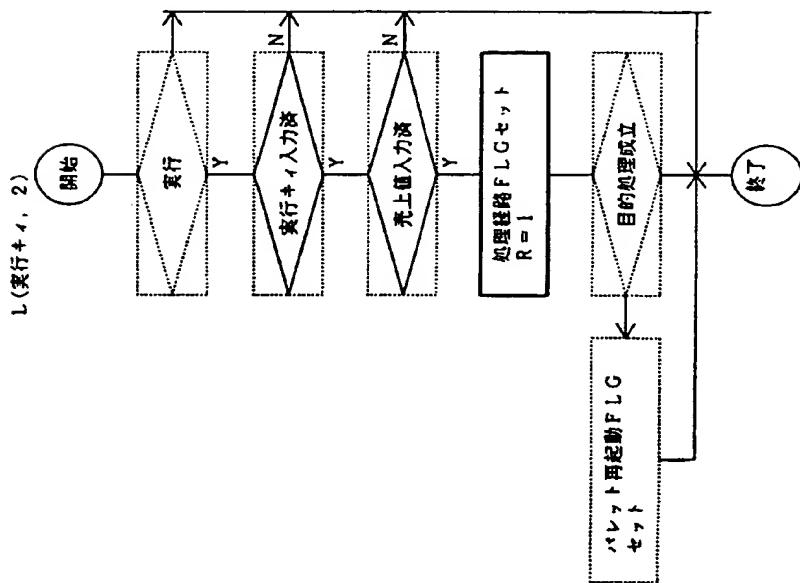


図 29

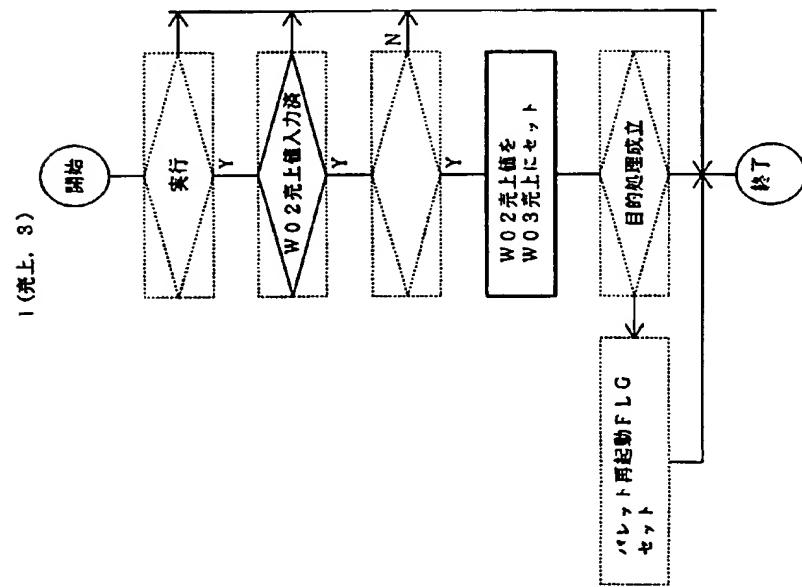
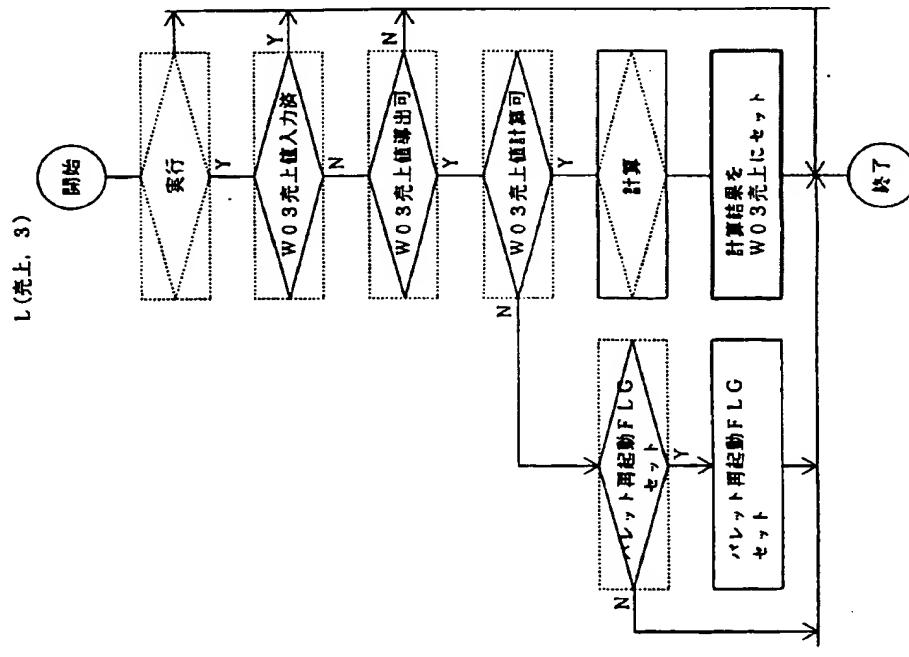


図 30

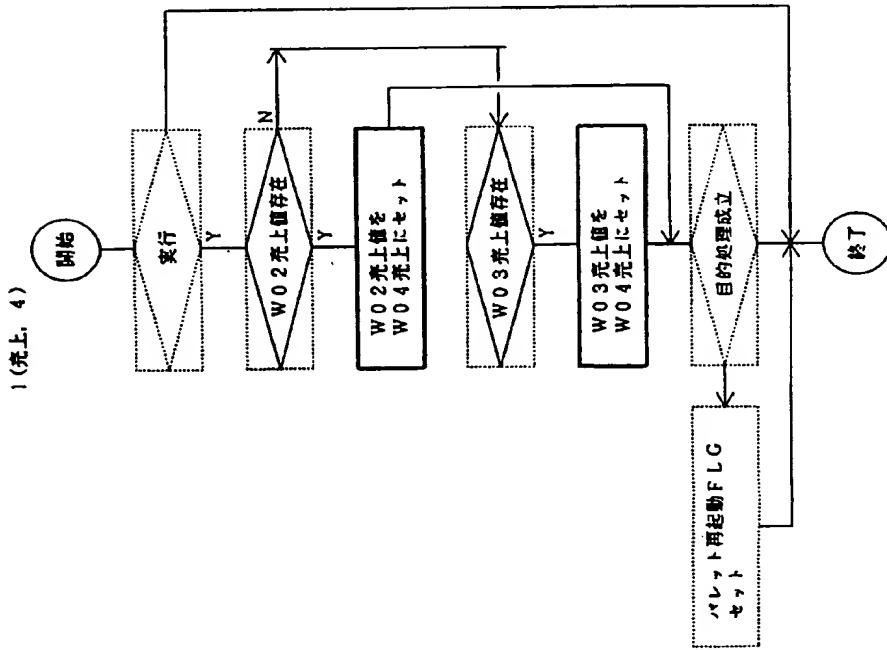
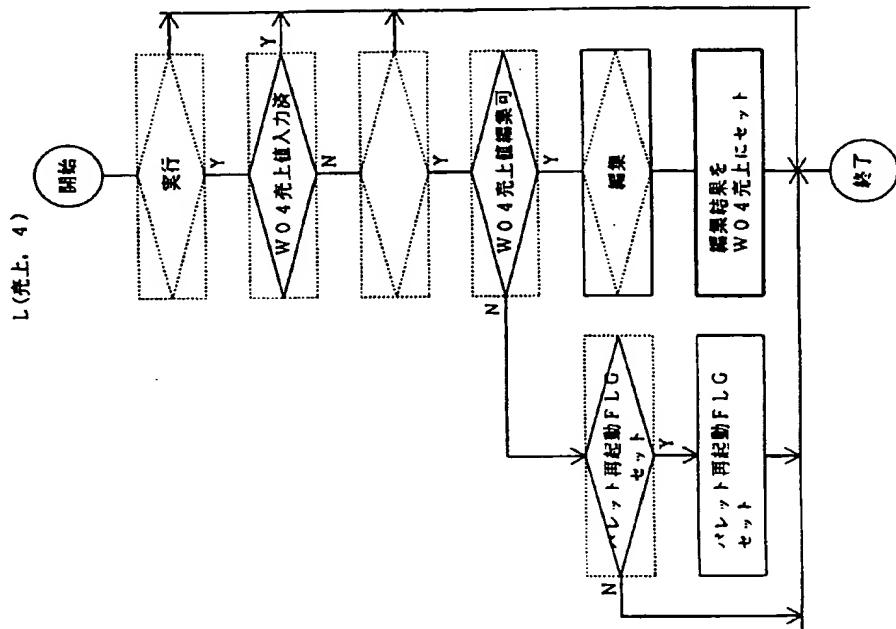


図 31

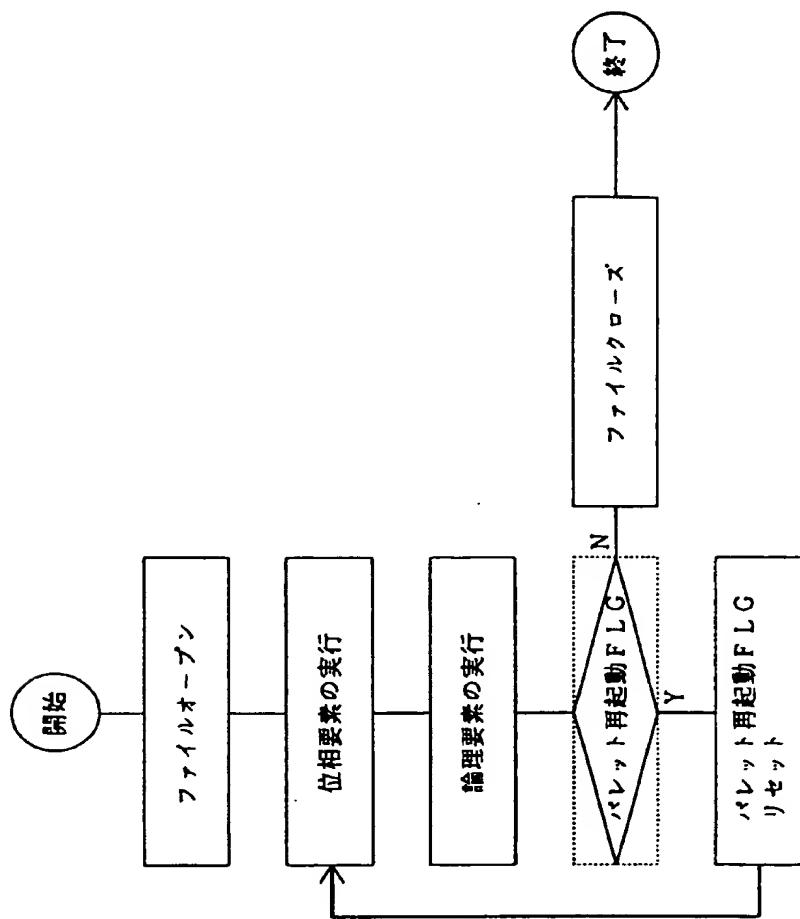
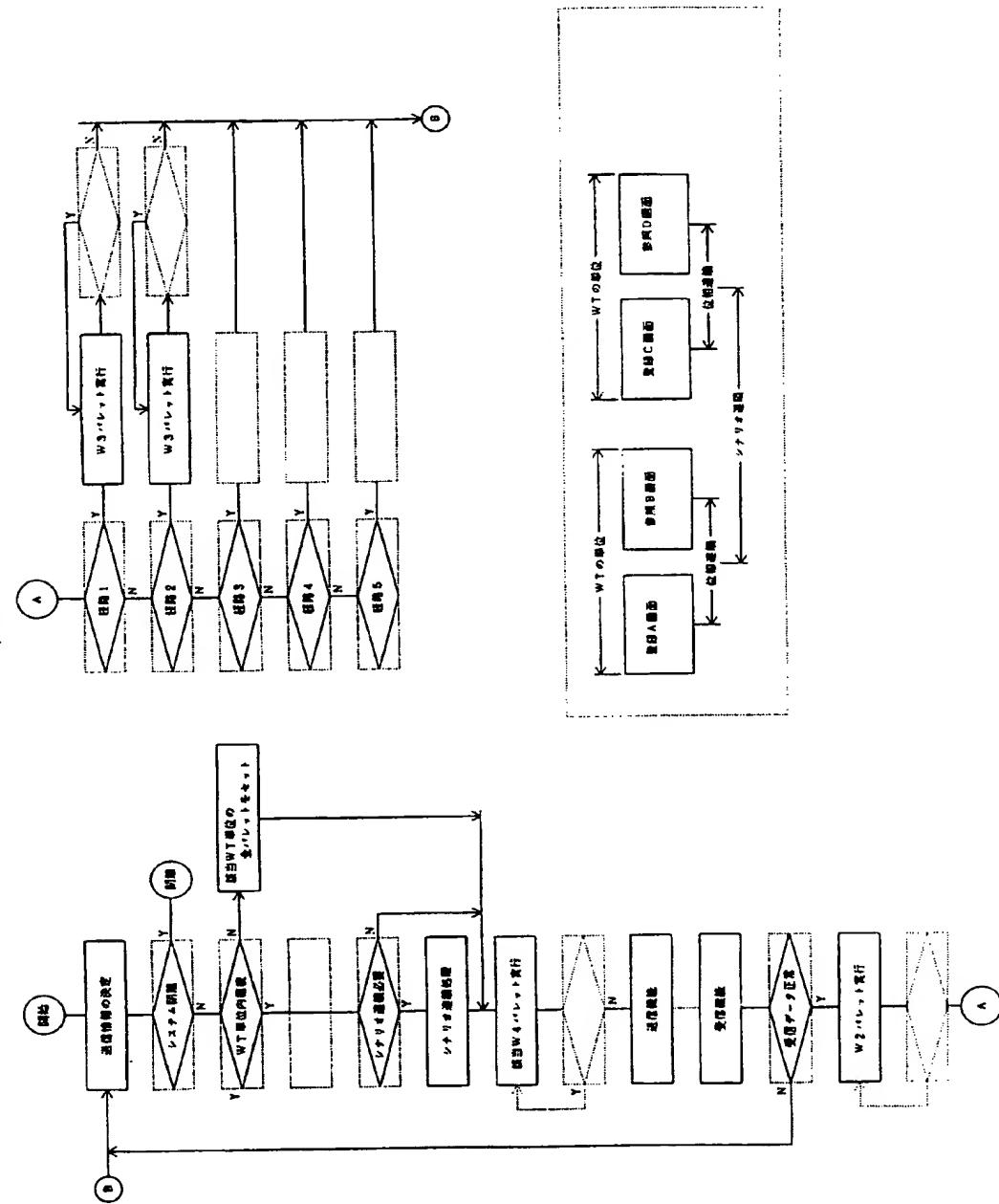
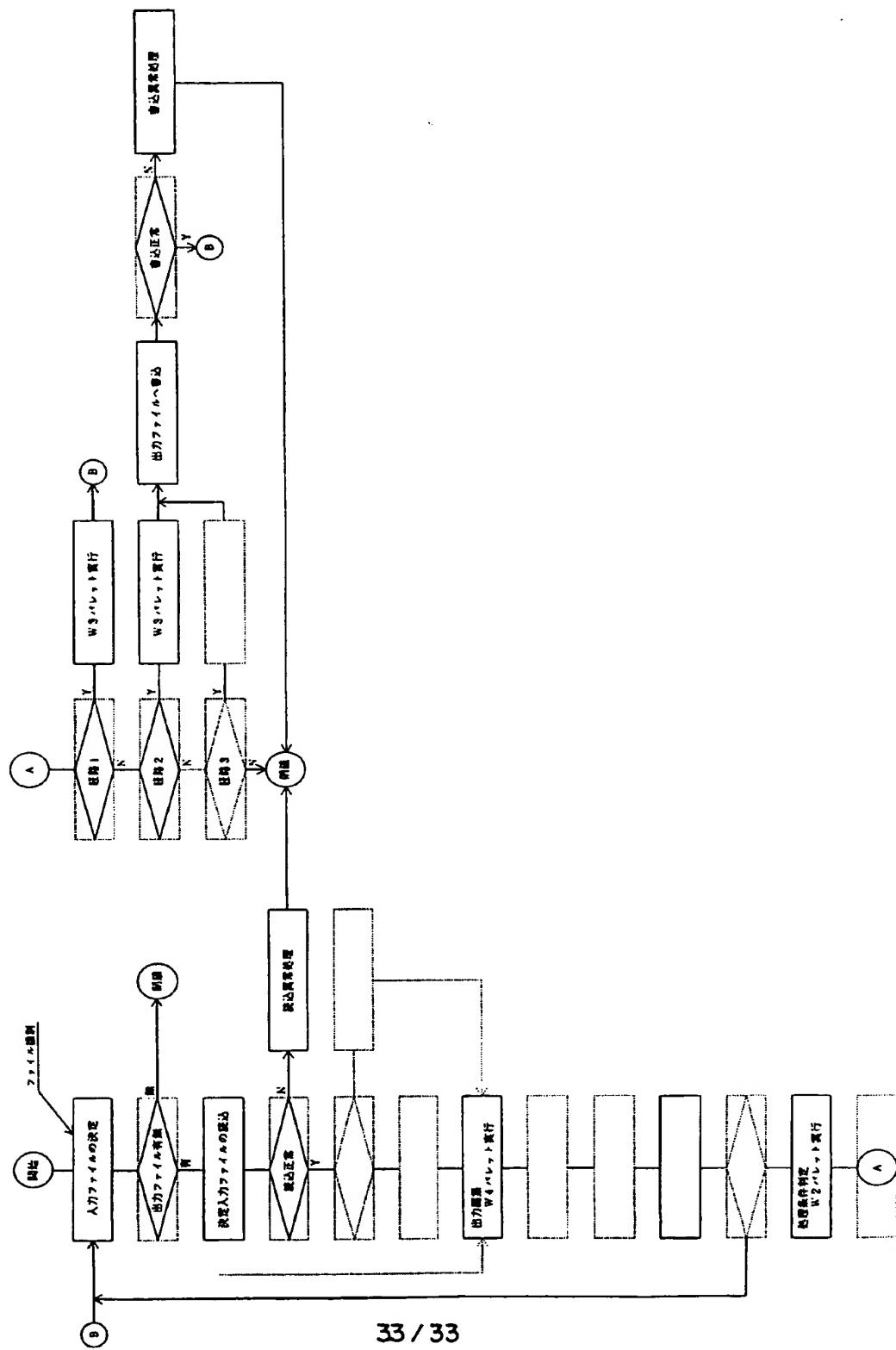


図32



33



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP96/03183

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl⁶ G06F9/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int. Cl⁶ G06F9/06

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1979 - 1997
Kokai Jitsuyo Shinan Koho	1972 - 1994
Toroku Jitsuyo Shinan Koho	1994 - 1997

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP, 6-332678, A (Institute of Software Scientifical Constructions Co., Ltd., K.K. East Japan Railway Information System), December 2, 1994 (02. 12. 94) (Family: none)	1 - 3
A	JP, 5-108319, A (Hitachi, Ltd.), April 30, 1993 (30. 04. 93) (Family: none)	1 - 3
A	JP, 4-137038, A (K.K. Business System Laboratory), May 12, 1992 (12. 05. 92) (Family: none)	1 - 3

Further documents are listed in the continuation of Box C. See patent family annex.

- * Special categories of cited documents:
- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed
- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search January 22, 1997 (22. 01. 97)	Date of mailing of the international search report February 4, 1997 (04. 02. 97)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

国際調査報告

国際出願番号 PCT/JP96/03183

A. 発明の属する分野の分類(国際特許分類(IPC))

Int. Cl⁶ G06F9/06

B. 調査を行った分野

調査を行った最小限資料(国際特許分類(IPC))

Int. Cl⁶ G06F9/06

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1979-1997年
日本国公開実用新案公報	1972-1994年
日本国登録実用新案公報	1994-1997年

国際調査で使用した電子データベース(データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP, 6-332678, A (株式会社 ソフトウェア構築科学研究所, 株式会社 ジェイアール東日本情報システム), 2. 12月, 1994 (02. 12. 94) (ファミリーなし)	1-3
A	JP, 5-108319, A (株式会社 日立製作所), 30. 4月, 1993 (30. 04. 93) (ファミリーなし)	1-3
A	JP, 4-137038, A (株式会社 ビジネスシステム研究所) 12. 5月, 1992 (12. 05. 92) (ファミリーなし)	1-3

 C欄の続きにも文献が列挙されている。 パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

- 「A」特に関連のある文献ではなく、一般的技術水準を示すもの
- 「E」先行文献ではあるが、国際出願日以後に公表されたものの
- 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献(理由を付す)
- 「O」口頭による開示、使用、展示等に言及する文献
- 「P」国際出願日前で、かつ優先権の主張の基礎となる出願
- 日の後に公表された文献
- 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
- 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
- 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
- 「&」同一パテントファミリー文献

国際調査を完了した日
22. 01. 97

国際調査報告の発送日 04.02.97

国際調査機関の名称及びあて先
日本国特許庁 (ISA/JP)
郵便番号100
東京都千代田区霞が関三丁目4番3号特許庁審査官(権限のある職員)
小川 譲 印
5B 9367
電話番号 03-3581-1101 内線 3545